```
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMM         MMM     AAAAAAAAA        CCCCCCCCCCCC  RRRRRRRRRRRR        OOOOOOOOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMMMMM   MMMMMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM   MMM   MMM  AAA         AAA  CCC              RRR         RRR  OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRRRRRRRRRRR     OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAAAAAAAAAAAAAA  CCC              RRR   RRR        OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA  CCC              RRR      RRR     OOO         OOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
MMM         MMM  AAA         AAA     CCCCCCCCCCCC  RRR         RRR     OOOOOOOOO
```

```
_$

Sy
---
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA

MA
MA
MA
MA

MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
MA
```

```
**FILE**ID**FINISH
```

```
FFFFFFFFFF    IIIIII    NN      NN    IIIIII      SSSSSSSS  HH      HH
FFFFFFFFFF    IIIIII    NN      NN    IIIIII      SSSSSSSS  HH      HH
FF              II      NN      NN      II      SS          HH      HH
FF              II      NN      NN      II      SS          HH      HH
FF              II      NNNN    NN      II      SS          HH      HH
FF              II      NNNN    NN      II      SS          HH      HH
FFFFFFFF        II      NN  NN  NN      II        SSSSSS    HHHHHHHHHH
FFFFFFFF        II      NN  NN  NN      II        SSSSSS    HHHHHHHHHH
FF              II      NN    NNNN      II              SS  HH      HH
FF              II      NN    NNNN      II              SS  HH      HH
FF              II      NN      NN      II              SS  HH      HH      ....
FF              II      NN      NN      II              SS  HH      HH      ....
FF            IIIIII    NN      NN    IIIIII    SSSSSSSS    HH      HH      ....
FF            IIIIII    NN      NN    IIIIII    SSSSSSSS    HH      HH      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

MAC
V04-

Table of contents

```
0000          .TITLE  MAC$FINISH      ROUTINES FOR FINISHING ASSEMBLY
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:     VAX MACRO ASSEMBLER OBJECT LIBRARY
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ; The VAX-11 MACRO assembler translates MACRO-32 source code into object
0000    ; modules for input to the VAX-11 LINKER.
0000    ;
0000    ; ENVIRONMENT:  USER MODE
0000    ;
0000    ; AUTHOR: Benn Schreiber, CREATION DATE: 25-AUG-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-004 MTR0035         Mike Rhodes             2-Aug-1983
0000    ;               Correct psect definition in symbol table listing when the
0000    ;               blank psect has been removed.
0000    ;
0000    ;       V03-003 MTR0033         Mike Rhodes             22-Apr-1983
0000    ;               Allow the removal of the blank psect if it is not referenced.
0000    ;
0000    ;       V03-002 MTR0031         Mike Rhodes             19-Apr-1983
0000    ;               Remove obsolete reference to $MAC_TIRCMDDEF macro.
0000    ;
0000    ;       V03-001 MTR0025         Mike Rhodes             8-Feb-1983
0000    ;               Modify routines PRT_PSECT_SYNOP, PRT_CROSS_REF, PRT_RUN_TIM,
0000    ;               PRT_MLB_STATS, CREF_TREE_OUT, and NEW_SBT_CK_PAGE to write
0000    ;               individual data records instead of having imbedded carriage
0000    ;               control.
```

```
0000    ;
0000    ;       V03.00  MTR0005         Mike Rhodes             15-Mar-1982
0000    ;               Modify routine MAC$DBG_PSECT to write a Position
0000    ;               Independent Data Reference (TIR$C_STO_PIDR) instead
0000    ;               of the Store Long Word (TIR$C_STO_LW).
0000    ;
0000    ;       V02.16  PCG0003         Peter George            07-May-1981
0000    ;               Create bug which causes global symbols that have
0000    ;               been suppressed from the symbol table to have their
0000    ;               global symbol records suppressed.
0000    ;
0000    ;       V02.15  PCG0001         Peter George            06-Feb-1981
0000    ;               Filter out creation of PSECT DST if ABS specified.
0000    ;
0000    ;       V02.14  CNH0035         Chris Hume              12-Jun-1980
0000    ;               Prevented extraneous symbols at the end of two column (31
0000    ;               character) symbol table output.
0000    ;
0000    ;       V01.13  RN0022          R. Newland      31-Oct-1979
0000    ;               Translate SYS$LP_LINES to set lines/page
0000    ;
0000    ;       V01.12  RN0013          R. Newland      27-Sep-1979
0000    ;               Use new symbols for PSECT options processing
0000    ;
0000    ;       V01.11  RN0011          R. Newland      11-Sep-1979
0000    ;               New librarian support
0000    ;
0000    ;       V01.10  RN0008          R. Newland      29-Aug-1979
0000    ;               31 character symbols
0000    ;
0000    ;       V01.09  RN0005          R. Newland      13-Aug-1979
0000    ;               Variable symbol name storage
0000    ;
0000    ;       V01.08  RN0002          R. Newland      01-Feb-1979
0000    ;               Changes for Source Update Merge, information messages
0000    ;               count and line number format
0000    ;
0000    ;
0000    ;       V01.06  BLS21346        B. Schreiber    29-DEC-1978
0000    ;               Remove restriction that symbol must be referenced
0000    ;               and not absolute to be output to debugger.
0000    ;--
```

```
                   0000             .SBTTL  DECLARATIONS
                   0000       ;
                   0000       ; INCLUDE FILES:
                   0000       ;
                   0000
                   0000       ;
                   0000       ; MACROS:
                   0000       ;
                   0000
                   0000             $MAC_SYMBLKDEF                         ;SYMBOL BLOCK DEFINITIONS
                   0000             $MAC_MLFDEF                            ; Define MLF offsets
                   0177             $MAC_GENVALDEF                         ;GENERAL VALUES
                   0177             $MAC_CTLFLGDEF                         ;CONTROL FLAGS
                   0177             $MAC_CRFLAGDEF                         ;DEFINE CREF CONTROL FLAGS
                   0177             $MAC_OBJCODDEF                         ;DEFINE OBJECT CODE COMMANDS
                   0000             $FABDEF                                ;DEFINE FAB OFFSETS
                   0000             $CRFDEF                                ; Define CRF offsets
                   0000
                   0000       ;
                   0000       ; LOCAL SYMBOLS
                   0000       ;
                   0000
         000000B8  0000       DBG$C_PSECT     =       184                  ;PSECT NAME (1+11+N-ENTRY LENGTH)
         000000BA  0000       DBG$C_SYMBOL    =       186                  ;SYMBOL NAME (1+7+N)
         000000BC  0000       DBG$C_MODULE    =       188                  ;MODULE NAME (1+7+N)
         000000BD  0000       DBG$C_MEND      =       189                  ;MODULE END (1+1)
         000000BE  0000       DBG$C_ROUTINE   =       190                  ;ROUTINE NAME (1+7+N)
         00000000  0000       DBG$C_LIT_DAT   =       0                    ;LITERAL DATA
         00000001  0000       DBG$C_REL_DAT   =       1                    ;RELOCATABLE DATA
         0000000C  0000       DBG$K_PSECT_LEN =       12                   ;CONSTANT LENGTH OF PSECT ENTRY
         00000008  0000       DBG$K_SYMBOL_LN =       8                    ;CONSTANT LENGTH OF SYMBOL ENTRY
         00000008  0000       DBG$K_MODULE_LN =       8                    ;CONSTANT LENGTH OF MODULE ENTRY
         00000002  0000       DBG$K_MEND_LEN  =       2                    ;CONSTANT LENGTH OF MODULE END ENTRY
         00000008  0000       DBG$K_ROUTIN_LN =       8                    ;CONSTANT LENGTH OF ROUTINE ENTRY
                   0000
                   0000       ;
                   0000       ; LOCAL DATA
                   0000       ;
                   0000
               00000000             .PSECT  MAC$RW_DATA,NOEXE,LONG
                   0000
                   0000       MAC$G_ERRBFDES::                             ;TO FAO THE ERROR LINE #'S
        00000000'  0000             .LONG   MAC$AB_LINE_END-MAC$AB_LST_END ;SIZE OF BUFFER
         00000000  0004             .LONG   0                              ;FILLED IN WITH ADDRESS
```

```
                          0008          .SBTTL  MAC$SORT_TABLE SORT SYMBOL TABLE ALPHABETICALLY
                          0008
                          0008  ;++
                          0008  ; FUNCTIONAL DESCRIPTION:
                          0008  ;
                          0008  ;       THIS ROUTINE WILL SORT THE SYMBOL TABLE INTO AN ALPHABETIC
                          0008  ;       LINKED LIST.
                          0008  ;
                          0008  ; INPUTS:
                          0008  ;
                          0008  ;       R9      SYMBOL HASH TABLE ADDRESS
                          0008  ;       R8      SYMBOL HASH TABLE SIZE
                          0008  ;
                          0008  ; OUTPUTS:
                          0008  ;
                          0008  ;       MAC$GL_LINK_PTR POINTER TO LINKED LIST OF SYMBOLS
                          0008  ;
                          0008  ;--
                          0008
                      00000000          .PSECT  MAC$RO_CODE_P3,NOWRT,GBL,LONG
                          0000
                          0000  MAC$SORT_TABLE::
       0000'CF    9F      0000          PUSHAB  W^MAC$GQ_RNT_SRT                ;STACK TIME BLOCK ADDRESS
  0000'CF    01   FB      0004          CALLS   #1,W^MAC$TIMER_ON               ;BEGIN TIMING SYMBOL TABLE SORT
                          0009  ;
                          0009  ; THE FIRST STEP IS TO REMOVE ALL EMPTY SUBLIST LISTHEADS FROM THE TABLE
                          0009  ;
          57    59  D0    0009          MOVL    R9,R7                           ;COPY TABLE ADDRESS
          56    57  D0    000C          MOVL    R7,R6                           ;...
                87  D5    000F  10$:    TSTL    (R7)+                           ;LIST HEAD EMPTY?
                04  13    0011          BEQL    20$                             ;IF EQL YES
       86    FC A7  D0    0013          MOVL    -4(R7),(R6)+                    ;NO--MOVE THE LISTHEAD ENTRY
             F5 58  F5    0017  20$:    SOBGTR  R8,10$                          ;SCAN ALL LISTHEADS
           0000'CF  D4    001A          CLRL    W^MAC$GL_LINK_PTR               ;ASSUME NO ENTRIES
    55    56    59  C3    001E          SUBL3   R9,R6,R5                        ;CALCULATE SIZE OF TABLE IN BYTES
                03  12    0022          BNEQ    SORT_LISTS                      ;BRANCH IF THERE ARE SOME ENTRIES
              00CE  31    0024          BRW     SORT_EXIT                       ;ELSE GO EXIT
                          0027  ;
                          0027  ; NEXT, WE SORT THE SUBLISTS SUCH THAT THE FIRST SYMBOL IN EACH SUBLIST
                          0027  ; IS LESS THAN THE FIRST SYMBOL OF THE SUCCEEDING SUBLIST.
                          0027  ;
                          0027  SORT_LISTS:
          58    59  D0    0027          MOVL    R9,R8                           ;COPY TABLE ADDRESS
          57    88  D0    002A  10$:    MOVL    (R8)+,R7                        ;GET ADDRESS OF NEXT SYMBOL
          56    58  D1    002D          CMPL    R8,R6                           ;ANY MORE LISTHEADS IN TABLE?
                30  13    0030          BEQL    MERGE_LISTS                     ;IF EQL NO
          55    68  D0    0032          MOVL    (R8),R5                         ;GET ADDRESS OF NEXT
       50    04 A7  9A    0035          MOVZBL  SYM$B_NAME(R7),R0               ; Get offset to name
    50    57    50  C3    0039          SUBL3   R0,R7,R0                        ; and form address of count/name
          52    80  9A    003D          MOVZBL  (R0)+,R2                        ; Get count and advance pointer to name
       51    04 A5  9A    0040          MOVZBL  SYM$B_NAME(R5),R1               ; Get offset to name
    51    55    51  C3    0044          SUBL3   R1,R5,R1                        ; and form address of count/name
          53    81  9A    0048          MOVZBL  (R1)+,R3                        ; Get count and advance pointer to name
61  53  00  60  52  2D    004B          CMPC5   R2,(R0),#0,R3,(R1)              ; Compare symbols
                D7  1F    0051          BLSSU   10$                             ;IF LSS THEN ORDER CORRECT
          68    57  D0    0053          MOVL    R7,(R8)                         ;INVERT LISTHEAD ENTRIES
          78    55  D0    0056          MOVL    R5,-(R8)                        ;...
```

```
                59    58    D1  0059               CMPL     R8,R9                     ;AT FRONT OF LISTHEAD TABLE?
                      CC    13  005C               BEQL     10$                       ;IF EQL YES
                      78    D5  005E               TSTL     -(R8)                     ;NO--BACKUP ONE ENTRY
                      C8    11  0060               BRB      10$
                                0062     ;
                                0062     ; NOW MERGE THE SUBLISTS INTO ONE MASTER SORTED LIST
                                0062     ;
                                0062     MERGE_LISTS:
         58      0000'CF    9E  0062               MOVAB    W^MAC$GL_LINK_PTR,R8      ;POINT TO MERGE LISTHEAD
                57    59    D0  0067               MOVL     R9,R7                     ;COPY LISTHEAD TABLE POINTER
                55    87    D0  006A     10$:      MOVL     (R7)+,R5                  ;GET ADDRESS OF NEXT SYMBOL
                      03    12  006D               BNEQ     12$                       ; If NEQ more in this sublist
                    007C    31  006F               BRW      70$
                                0072     12$:
       2C 09 A5       06    E0  0072               BBS      #SYM$V_LOCAL,SYM$W_FLAG(R5),20$ ;BR IF LOCAL SYMBOL
       0A 09 A5       00    E0  0077               BBS      #SYM$V_DEF,SYM$W_FLAG(R5),15$ ;BRANCH IF DEFINED
       05 09 A5       03    E0  007C               BBS      #SYM$V_EXTRN,SYM$W_FLAG(R5),15$ ;BRANCH IF DECLARED EXTERNAL
       21 09 A5       07    E1  0081               BBC      #SYM$V_REF,SYM$W_FLAG(R5),30$ ;BRANCH IF NOT REFERENCED
                                0086                                                  ;(SYMBOL WAS USED IN .NTYPE DIRECTIVE
                                0086                                                  ;AND WAS NOT REF. OTHERWISE)
             0000'CF        D6  0086     15$:      INCL     W^MAC$GL_SYM_NLOC         ;COUNT NON-LOCAL SYMBOL
                68    55    D0  008A               MOVL     R5,SYM$L_LINK(R8)         ; Link new symbol to old last
                58    55    D0  008D               MOVL     R5,R8                     ;MAKE NEW LAST SYMBOL
                                0090     ;
                                0090     ; IF .ENABLE GLOBAL AND SYMBOL IS UNDEFINED, DECLARE IT EXTERNAL NOW.
                                0090     ;
     10 00000005'EF         E9  0090               BLBC     L^ENB$G_GLOBAL+SYM$L_VAL,30$ ;BRANCH IF DISABLE GLOBAL
       0B 09 A5       00    E0  0097               BBS      #SYM$V_DEF,SYM$W_FLAG(R5),30$ ;BRANCH IF SYMBOL DEFINED
       06 09 A5       03    E3  009C               BBCS     #SYM$V_EXTRN,SYM$W_FLAG(R5),30$ ;DECLARE SYMBOL EXTERNAL
                      04    11  00A1               BRB      30$                       ;GO AHEAD
             0000'CF        D6  00A3     20$:      INCL     W^MAC$GL_SYM_LOCL         ;COUNT A LOCAL SYMBOL
                77    65    D0  00A7     30$:      MOVL     SYM$L_LINK(R5),-(R7)      ; Remove symbol from list
                      02    12  00AA               BNEQ     40$                       ;IF NEQ THEN MORE IN LIST
                      87    D5  00AC               TSTL     (R7)+                     ;ADVANCE LISTHEAD POINTER
                56    57    D1  00AE     40$:      CMPL     R7,R6                     ;ANY MORE SUBLISTS TO CONSIDER?
                      40    13  00B1               BEQL     80$                       ;IF EQL NO
                7E    57    7D  00B3               MOVQ     R7,-(SP)                  ;SAVE MERGE PARMS (R7/R8)
                58    87    D0  00B6     50$:      MOVL     (R7)+,R8                  ;GET ADDRESS OF FIRST SUBLIST ENTRY
                56    57    D1  00B9               CMPL     R7,R6                     ;ANY MORE SUBLISTS TO CONSIDER?
                      2A    13  00BC               BEQL     60$                       ;IF EQL NO
                55    67    D0  00BE               MOVL     (R7),R5                   ;YES--GET ADDRESS OF FIRST SUBLIST ENTRY
          50   04 A8        9A  00C1               MOVZBL   SYM$B_NAME(R8),R0         ; Get offset to first name
         50     58    50    C3  00C5               SUBL3    R0,R8,R0                  ; and form address of count/name
                52    80    9A  00C9               MOVZBL   (R0)+,R2                  ; Get count and advance pointer to name
          51   04 A5        9A  00CC               MOVZBL   SYM$B_NAME(R5),R1         ; Get offset to second name
         51     55    51    C3  00D0               SUBL3    R1,R5,R1                  ; and form address of count/name
                53    81    9A  00D4               MOVZBL   (R1)+,R3                  ; Get count and advance pointer to name
 61    53    00    60    52    2D  00D7               CMPC5    R2,(R0),#0,R3,(R1)        ; Compare symbols
                      09    1F  00DD               BLSSU    60$                       ;IF LSS ORDER CORRECT
                67    58    D0  00DF               MOVL     R8,(R7)                   ;SWAP SUBLISTS
             FC A7    55    D0  00E2               MOVL     R5,-4(R7)                 ;...
                      CE    11  00E6               BRB      50$                       ;CONTINUE
                57    8E    7D  00E8     60$:      MOVQ     (SP)+,R7                  ;RESTORE MERGE PARAMETERS (R7/R8)
                    FF7C    31  00EB     65$:      BRW      10$                       ; Continue merge
                56    57    D1  00EE     70$:      CMPL     R7,R6                     ;ANY MORE SUBLISTS TO CONSIDER?
                      F8    12  00F1               BNEQ     65$                       ; If NEQ yes
                      68    D4  00F3     80$:      CLRL     SYM$L_LINK(R8)            ; Clear forward link of last entry
```

```
                              00F5    260 SORT_EXIT:
                 0000'CF   9F 00F5    261         PUSHAB  W^MAC$GQ_RNT_SRT         ;STACK TIMING BLOCK ADDRESS
         0000'CF      01   FB 00F9    262         CALLS   #1,W^MAC$TIMER_OFF       ;STOP TIMING SYMBOL TABLE SORT
                           05 00FE    263         RSB
```

```
                         00FF                   .SBTTL  MAC$FINISH_ASM OUTPUT TRACEBACK, DEBUG INFO
                         00FF
                         00FF   ;++
                         00FF   ; FUNCTIONAL DESCRIPTION:
                         00FF   ;
                         00FF   ;               THIS ROUTINE IS CALLED AT THE END OF THE ASSEMBLY TO FINISH
                         00FF   ;               THINGS UP.  IF ENABLED, TRACEBACK AND DEBUG INFORMATION IS
                         00FF   ;               OUTPUT TO THE OBJECT FILE.  THE SYMBOL TABLE LISTING IS PRINTED.
                         00FF   ;
                         00FF   ;--
                         00FF
                         00FF   MAC$FINISH_ASM::
           0380 8F  BB   00FF                   PUSHR   #^M<R7,R8,R9>                   ;SAVE REGISTERS
                57  D4   0103                   CLRL    R7                              ;RESET STACK POINTER
        03 6B   15  E0   0105                   BBS     #FLG$V_OBJXST,(R11),5$          ; Branch if there is an object file
              0084  31   0109                   BRW     85$                             ; No object file
                         010C   5$:
    7D 00000005'EF  E9   010C                   BLBC    L^ENB$G_TRACEBACK+SYM$L_VAL,85$ ; Branch if disable traceback
      0000'CF   05  9A   0113                   MOVZBL  #OBJ$C_TBT,W^MAC$GL_RECTYP ;TRACEBACK RECORD TYPE
              FEE5' 30   0118                   BSBW    MAC$WRTOBJ                      ;WRITE OUT LAST TIR RECORD
      55   0000'CF  9E   011B                   MOVAB   W^MAC$AB_TITLE,R5               ;POINT TO TITLE STRING
                65  95   0120                   TSTB    (R5)                            ;WAS TITLE SUPPLIED?
                07  12   0122                   BNEQ    10$                             ;IF NEQ YES--GO USE IT
  55   00000000'EF  9E   0124                   MOVAB   L^MAC$AB_DEF_TITL,R5            ;POINT TO DEFAULT TITLE STRING
      50   07   65  81   012B   10$:            ADDB3   (R5),#DBG$K_MODULE_LN-1,R0 ;FIGURE LENGTH OF ENTRY
              FECE' 30   012F                   BSBW    MAC$STOIM                       ;STORE INTO OBJECT CODE
        50   BC 8F  9A   0132                   MOVZBL  #DBG$C_MODULE,R0                ;MODULE NAME
              FEC7' 30   0136                   BSBW    MAC$STOIM                       ;STORE IT
           54   05  9A   0139                   MOVZBL  #5,R4                           ;SET A LOOP COUNT
                50  D4   013C   20$:            CLRL    R0                              ;STRUCTURE LEVEL AND 'MACRO?(?)
              FEBF' 30   013E                   BSBW    MAC$STOIM
             F8 54  F5   0141                   SOBGTR  R4,20$                          ;DO 5
           54   85  9A   0144                   MOVZBL  (R5)+,R4                        ;GET CHARACTER COUNT
           50   54  D0   0147                   MOVL    R4,R0                           ;COPY TO STORE IT
              FEB3' 30   014A                   BSBW    MAC$STOIM
           50   85  9A   014D   40$:            MOVZBL  (R5)+,R0                        ;GET CHAR OF TITLE
              FEAD' 30   0150                   BSBW    MAC$STOIM                       ;STORE IMMEDIATE
             F7 54  F5   0153                   SOBGTR  R4,40$
                         0156   ;
                         0156   ; SEARCH SYMBOL TABLE AND EMIT ENTRY POINTS TO THE DEBUGGER
                         0156   ;
      56   0000'CF  D0   0156                   MOVL    W^MAC$GL_LINK_PTR+SYM$L_LINK,R6 ; Point to linked symbol list
                2B  13   015B                   BEQL    80$                             ;IF EQL THERE IS NO LIST
     21 09 A6   09  E1   015D   60$:            BBC     #SYM$V_EPT,SYM$W_FLAG(R6),70$ ;BRANCH IF NOT EPT
        50   04 A6  9A   0162                   MOVZBL  SYM$B_NAME(R6),R0               ; Get offset to name
      50   56   50  C3   0166                   SUBL3   R0,R6,R0                        ; Form address of count/name
                07  81   016A                   ADDB3   #DBG$K_SYMBOL_LN-1,-            ; Figure length of entry
           50   60       016C                           (R0),R0                         ; putting result in R0
              FE8F' 30   016E                   BSBW    MAC$STOIM
        50   BE 8F  9A   0171                   MOVZBL  #DBG$C_ROUTINE,R0               ;TYPE
              FE88' 30   0175                   BSBW    MAC$STOIM
                50  D4   0178                   CLRL    R0                              ;ZERO BYTE
              FE83' 30   017A                   BSBW    MAC$STOIM
              06E6  30   017D                   BSBW    MAC$DBG_VAL_OUT                 ;OUTPUT ADDRESS
              06D1  30   0180                   BSBW    MAC$DBG_NAM_OUT                 ;OUTPUT NAME
           56   66  D0   0183   70$:            MOVL    SYM$L_LINK(R6),R6               ; Link to next symbol
                D5  12   0186                   BNEQ    60$                             ;IF THERE IS ONE, GO PROCESS IT
```

```
                        0188    ;
                        0188    ; OUPTUT LABLES AND REFERENCED SYMBOLS FOR DEBUGGER
                        0188    ;
   0000'CF    04    9A  0188    80$:    MOVZBL  #OBJ$C_DBG,W^MAC$GL_RECTYP ;SET DEBUG RECORD TYPE
           FE70'    30  018D            BSBW    MAC$WRTOBJ                 ;WRITE OUT THE TRACEBACK RECORDS
    56   0000'CF    D0  0190    85$:    MOVL    W^MAC$GL_LINK_PTR,R6       ;POINT TO SYMBOL LIST AGAIN
              64    13  0195            BEQL    150$                       ;IF EQL THERE IS NO LIST
                        0197    ;
                        0197    ; BACK HERE FOR EACH SYMBOL IN THE LIST
                        0197    ;
      53   09 A6    3C  0197    90$:    MOVZWL  SYM$W_FLAG(R6),R3          ;GET FLAGS FOR SYMBOL
      57 53   05    E1  019B            BBC     #SYM$V_DEBUG,R3,140$       ;BRANCH IF NO DEBUG ATTRIBUTE
      53 53   00    E1  019F            BBC     #SYM$V_DEF,R3,140$         ;BRANCH IF NOT DEFINED
                        01A3    ;
                        01A3    ; SYMBOL HAS DEBUG ATTRIBUTE AND IS DEFINED.
                        01A3    ;
  09 A6    0400 8F  A8  01A3    100$:   BISW2   #SYM$M_ODBG,SYM$W_FLAG(R6) ;SET DEBUG OUTPUT FLAG FOR SYMBOL
      49 53   09    E0  01A9            BBS     #SYM$V_EPT,R3,140$         ;BRANCH IF THIS IS AN ENTRY POINT
      45 6B   15    E1  01AD            BBC     #FLG$V_OBJXST,(R11),140$ ;BRANCH IF THERE IS NO OBJECT FILE
 3E 00000005'EF     E9  01B1            BLBC    L^ENB$G_TRACEBACK+SYM$L_VAL,140$ ;BRANCH IF DISABLE TRACEBACK
      50   04 A6    9A  01B8            MOVZBL  SYM$B_NAME(R6),R0          ; Get offset to name
    50   56   50    C3  01BC            SUBL3   R0,R6,R0                   ; Form address of count/name
              07    81  01C0            ADDB3   #DBG$K_SYMBOL_LN-1,-       ; Figure length of entry
         50   60        01C2                    (R0),R0                    ; putting result in R0
           FE39'    30  01C4            BSBW    MAC$STOIM
      50   BA 8F    9A  01C7            MOVZBL  #DBG$C_SYMBOL,R0           ;ASSUME A LABEL
           FE32'    30  01CB            BSBW    MAC$STOIM                  ;STORE INTO OBJECT CODE
      0B 52   04    E0  01CE            BBS     #SYM$V_ABS,R2,110$         ;BRANCH IF SYMBOL IS ABSOLUTE
                        01D2    ;
                        01D2    ; SYMBOL IS RELOCATABLE
                        01D2    ;
         50   01    9A  01D2            MOVZBL  #DBG$C_REL_DAT,R0          ;RELOCATABLE DATA
           FE28'    30  01D5            BSBW    MAC$STOIM
            068B    30  01D8            BSBW    MAC$DBG_VAL_OUT            ;OUTPUT SYMBOL ADDRESS
              16    11  01DB            BRB     130$
                        01DD    ;
                        01DD    ; SYMBOL IS ABSOLUTE
                        01DD    ;
         50   00    9A  01DD    110$:   MOVZBL  #DBG$C_LIT_DAT,R0          ;LITERAL DATA
           FE1D'    30  01E0            BSBW    MAC$STOIM
      55   05 A6    9E  01E3            MOVAB   SYM$L_VAL(R6),R5           ;POINT TO SYMBOL VALUE
         54   04    9A  01E7            MOVZBL  #4,R4                      ;LOOP COUNT
         50   85    90  01EA    120$:   MOVB    (R5)+,R0                   ;GET BYTE OF VALUE
           FE10'    30  01ED            BSBW    MAC$STOIM                  ;STORE INTO OBJECT CODE
           F7 54    F5  01F0            SOBGTR  R4,120$
                        01F3    ;
                        01F3    ; OUTPUT SYMBOL NAME
                        01F3    ;
            065E    30  01F3    130$:   BSBW    MAC$DBG_NAM_OUT            ;SEND NAME TO OBJECT FILE
         56   66    D0  01F6    140$:   MOVL    SYM$L_LINK(R6),R6          ; Link to next symbol
              9C    12  01F9            BNEQ    90$                        ;BRANCH IF THERE IS ONE
      40 6B   15    E1  01FB    150$:   BBC     #FLG$V_OBJXST,(R11),PRINT_SYM_TABLE ;SKIP IF NO OBJECT FILE
 39 00000005'EF     E9  01FF            BLBC    L^ENB$G_TRACEBACK+SYM$L_VAL,PRINT_SYM_TABLE ;SKIP IF DISABLE TRACEBA
   0000'CF    05    9A  0206            MOVZBL  #OBJ$C_TBT,W^MAC$GL_RECTYP ;SET FOR TRACEBACK RECORD TYPE
           FDF2'    30  020B            BSBW    MAC$WRTOBJ                 ;WRITE OUT DEBUG RECORD
                        020E    ;
                        020E    ; OUTPUT PSECT NAMES AND END OF MODULE
```

```
                     020E      :
   56   0000'CF  9E  020E                MOVAB   W^PSECT$BLANK,R6             ;BLANK PSECT
03 09 A6     07  E1  0213                BBC     #SYM$V_REF,SYM$W_FLAG(R6),155$ ;HAS IT BEEN REF'D?
           0677  30  0218                BSBW    MAC$DBG_PSECT                ;YES, OUTPUT INFORMATION
   56   0000'CF  D0  021B      155$:     MOVL    W^MAC$GL_PSC_LIST,R6         ;GET FIRST PSECT ADDRESS
             0D  13  0220                BEQL    170$                         ;IF EQL NONE
03 0D A6     03  E1  0222      160$:     BBC     #PSC$V_REL, -                ;SUPPRESS ABS PSECTS
                     0227                        PSC$W_OPTIONS(R6),165$
           0668  30  0227                BSBW    MAC$DBG_PSECT                ;OUPTUT INFO
        56   66  D0  022A      165$:     MOVL    SYM$L_LINK(R6),R6            ; Link to next symbol
             F3  12  022D                BNEQ    160$                         ;IF NEQ THERE IS MORE
        50   01  9A  022F      170$:     MOVZBL  #1,R0                        ;STORE LENGTH
          FDCB'  30  0232                BSBW    MAC$STOIM
     50   BD 8F  9A  0235                MOVZBL  #DBG$C_MEND,R0               ;MODULE END
          FDC4'  30  0239                BSBW    MAC$STOIM
          FDC1'  30  023C                BSBW    MAC$WRTOBJ                   ;WRITE OUT FINAL TRACEBACK RECORD
```

```
                              023F          .SBTTL  PRINT SYMBOL TABLE IN ALPHABETICAL ORDER
                              023F
                              023F  PRINT_SYM_TABLE:
                     5A   DD  023F          PUSHL   R10                     ;SAVE INT. BUFFER POINTER
            03 6B    09   E0  0241          BBS     #FLG$V_LSTXST,(R11),10$ ;BRANCH IF THERE IS LISTING FILE
                   0425   31  0245          BRW     PRT_ERR_SUM             ;NO--GO SEE ABOUT ERROR SUMMARY
                0000'CF   9F  0248  10$:    PUSHAB  W^MAC$GQ_RNT_P2         ;STACK TIMING BLOCK ADDRESS
          0000'CF    01   FB  024C          CALLS   #1,W^MAC$TIMER_OFF      ;STOP TIMING PASS 2 NOW
                0000'CF   9F  0251          PUSHAB  W^MAC$GQ_RNT_SYO        ;STACK TIMING BLOCK ADDRESS
          0000'CF    01   FB  0255          CALLS   #1,W^MAC$TIMER_ON       ;BEGIN TIMING SYMBOL TABLE OUTPUT
          0000'CF    01   9A  025A          MOVZBL  #1,W^MAC$GL_LIST_LVL    ;FORCE SYMBOL TABLE TO LIST
                     57   D4  025F          CLRL    R7                      ;RESET STACK POINTER
       50   00000000'EF   9E  0261          MOVAB   L^MAC$AB_STB_MSG,R0     ;POINT TO ''SYMBOL TABLE'' STRING
               51    80   9A  0268          MOVZBL  (R0)+,R1                ;GET ITS LENGTH
 0048 8F   20  60    51   2C  026B          MOVC5   R1,(R0),#^A/ /, -       ; Copy into subtitle buffer
         00000000'EF          0272
                              0277                          #LST$K_TITLE_SIZ+SYM$K_MAXLEN+1, -
                              0277                          L^MAC$AB_SBT_IDNT       ;(OVER IDENT AND SUBTITLE)
                0000'CF   D4  0277          CLRL    W^MAC$GL_DIRFLG         ;CLEAR NEW PAGE OUTPUT FLAG
          5A    0000'CF   D0  027B          MOVL    W^MAC$GL_LINK_PTR+SYM$L_LINK,R10 ; Point to symbol list
                     03   12  028C          BNEQ    20$                     ;IF NEQ GO PRINT SYMBOLS
                   00C7   31  0282          BRW     PRT_SYM_END             ;ELSE ALL DONE HERE
                0000'CF   D6  0285  20$:    INCL    W^MAC$GL_DIRFLG         ;SET NEW PAGE OUTPUT FLAG
                              0289  PRT_SYM_LOOP:
                  FD74'   30  0289          BSBW    MAC$LST_PAG_HDR         ;SKIP TO NEW PAGE
               58    5A   D0  028C          MOVL    R10,R8                  ;POINT TO SYMBOL
          55    0000'CF   D0  028F          MOVL    W^MAC$GL_LN_PAGE,R5     ; Lines of symbols/page
                     58   D5  0294  10$:    TSTL    R8                      ;END OF LIST?
                     14   13  0296          BEQL    20$                     ;IF EQL YES
                     55   D5  0298          TSTL    R5                      ;NO--END OF PAGE?
                     10   15  029A          BLEQ    20$                     ;IF LEQ YES
         05 09 A8    0E   E1  029C          BBC     #SYM$V_SUPR, -          ;BRANCH IF NOT SUPPRESSED SYMBOL
                              02A1                  SYM$W_FLAG(R8),15$
               58    68   D0  02A1          MOVL    SYM$L_LINK(R8),R8       ;LINK TO NEXT SYMBOL
                     EE   11  02A4          BRB     10$                     ;TRY AGAIN
               58    68   D0  02A6  15$:    MOVL    SYM$L_LINK(R8),R8       ;NOT SUPPRESSED, LINK TO NEXT SYMBOL
                  E8 55   F4  02A9          SOBGEQ  R5,10$                  ;COUNT AND LOOP
          55    0000'CF   D0  02AC  20$:    MOVL    W^MAC$GL_LN_PAGE,R5     ; Lines of symbols/page
            1B 6B    2A   E0  02B1          BBS     #FLG$V_SYM2COL,(R11),40$ ; Branch if two column listing
               59    58   D0  02B5          MOVL    R8,R9                   ;R8 POINTS TO FIRST SYMBOL IN
                              02B8                                          ;SECOND COLUMN...NOW FIND FIRST
                              02B8                                          ;FOR THIRD COLUMN
                     59   D5  02B8  30$:    TSTL    R9                      ;END OF LIST?
                     14   13  02BA          BEQL    40$                     ;IF EQL YES
                     55   D5  02BC          TSTL    R5                      ;NO--END OF PAGE?
                     10   15  02BE          BLEQ    40$                     ;IF LEQ YES
         05 09 A9    0E   E1  02C0          BBC     #SYM$V_SUPR, -          ;BRANCH IF NOT SUPPRESSED SYMBOL
                              02C5                  SYM$W_FLAG(R9),35$
               59    69   D0  02C5          MOVL    SYM$L_LINK(R9),R9       ;LINK TO NEXT SYMBOL
                     EE   11  02C8          BRB     30$                     ;TRY AGAIN
               59    69   D0  02CA  35$:    MOVL    SYM$L_LINK(R9),R9       ;NOT SUPPRESSED, LINK TO NEXT SYMBOL
                  E8 55   F4  02CD          SOBGEQ  R5,30$                  ;COUNT AND LOOP
                              02D0  ;
                              02D0  ; NOW R10 POINTS TO FIRST SYMBOL OF FIRST COLUMN
                              02D0  ;         R8 POINTS TO FIRST SYMBOL OF SECOND COLUMN
                              02D0  ;         R9 POINTS TO FIRST SYMBOL OF THIRD COLUMN
                              02D0  ;
```

```
             0000'CF   DD  02D0  40$:    PUSHL   W^MAC$GL_LN_PAGE          ; Stack symbols/page
                07DC   30  02D4  50$:    BSBW    MAC$LIST_INIT             ;INIT LISTING BUFFER AND POINTER
             56   5A   D0  02D7  53$:    MOVL    R10,R6                    ;POINT TO FIRST COLUMN SYMBOL
                  10   13  02DA          BEQL    60$                       ;IF EQL THERE IS NONE
       05 09 AA   0E   E1  02DC          BBC     #SYM$V_SUPR, -            ;BRANCH IF NOT SUPPRESSED SYMBOL
                           02E1                  SYM$W_FLAG(R10),57$
             5A   6A   D0  02E1          MOVL    SYM$L_LINK(R10),R10       ;LINK TO NEXT SYMBOL
                  F1   11  02E4          BRB     53$                       ;TRY AGAIN
             5A   6A   D0  02E6  57$:    MOVL    SYM$L_LINK(R10),R10       ;NOT SUPPRESSED, LINK TO NEXT SYMBOL
                06E1   30  02E9          BSBW    MAC$PRT_SYM_INF           ;PRINT THE SYMBOL INFORMATION
             56   58   D0  02EC  60$:    MOVL    R8,R6                     ;POINT TO SECOND COLUMN SYMBOL
                  10   13  02EF          BEQL    70$                       ;IF EQL THERE IS NONE
       05 09 A8   0E   E1  02F1          BBC     #SYM$V_SUPR, -            ;BRANCH IF NOT SUPPRESSED SYMBOL
                           02F6                  SYM$W_FLAG(R8),65$
             58   68   D0  02F6          MOVL    SYM$L_LINK(R8),R8         ;LINK TO NEXT SYMBOL
                  F1   11  02F9          BRB     60$                       ;TRY AGAIN
             58   68   D0  02FB  65$:    MOVL    SYM$L_LINK(R8),R8         ;NOT SUPPRESSED, LINK TO NEXT SYMBOL
                06CC   30  02FE          BSBW    MAC$PRT_SYM_INF           ;PRINT SYMBOL INFORMATION
          15 6B   2A   E0  0301  70$:    BBS     #FLG$V_SYM2COL,(R11),80$        ; Branch if two column listing
             56   59   D0  0305  73$:    MOVL    R9,R6                     ;POINT TO THIRD COLUMN SYMBOL
                  10   13  0308          BEQL    80$                       ;IF EQL THERE IS NONE
       05 09 A9   0E   E1  030A          BBC     #SYM$V_SUPR, -            ;BRANCH IF NOT SUPPRESSED SYMBOL
                           030F                  SYM$W_FLAG(R9),77$
             59   69   D0  030F          MOVL    SYM$L_LINK(R9),R9         ;LINK TO NEXT SYMBOL
                  F1   11  0312          BRB     73$                       ;TRY AGAIN
             59   69   D0  0314  77$:    MOVL    SYM$L_LINK(R9),R9         ;NOT SUPPRESSED, LINK TO NEXT SYMBOL
                06B3   30  0317          BSBW    MAC$PRT_SYM_INF           ;PRINT SYMBOL INFORMATION
00000000'8F  00000000'EF   D1  031A  80$:    CMPL    L^MAC$GL_LIST_PTR,#MAC$AB_LST_END ;LINE THERE?
                  11   1B  0325          BLEQU   90$                       ;IF LEQ NO
         00000000'8F   C3  0327          SUBL3   #MAC$AB_LINEBF,-          ;YES--FIGURE LINE LENGTH
         00000000'EF       032D                        L^MAC$GL_LIST_PTR,-
             0000'CF       0332                        W^MAC$GL_LINELN
               FCC8'   30  0335          BSBW    MAC$WRTLST                ;WRITE IT OUT
               99 6E   F5  0338  90$:    SOBGTR  (SP),50$                  ;LOOP FOR A PAGE
                  8E   D5  033B          TSTL    (SP)+                     ;CLEAN STACK
          03 6B   2A   E1  033D          BBC     #FLG$V_SYM2COL,(R11),95$        ; Branch if not two column listing
             59   58   D0  0341          MOVL    R8,R9                     ; Get next symbol to print
                           0344  95$:
             5A   59   D0  0344          MOVL    R9,R10                    ;POINT FOR NEXT PAGE
                  03   13  0347          BEQL    100$                      ;IF EQL ALL DONE
                FF3D   31  0349          BRW     PRT_SYM_LOOP              ;ELSE CONTINUE
                           034C  100$:
                           034C  PRT_SYM_END:
             0000'CF   9F  034C          PUSHAB  W^MAC$GQ_RNT_SYO          ;STACK TIMING BLOCK ADDRESS
        0000'CF   01   FB  0350          CALLS   #1,W^MAC$TIMER_OFF        ;FINISH TIMING SYMBOL TABLE OUTPUT
```

```
                               0355                .SBTTL  PRINT PSECT SYNOPSIS
                               0355
                               0355     PRT_PSECT_SYNOP:
             0000'CF     9F    0355            PUSHAB  W^MAC$GQ_RNT_PSY          ;STACK TIMING BLOCK ADDRESS
        0000'CF   01     FB    0359            CALLS   #1,W^MAC$TIMER_ON         ;START TIMING PSECT SYNOPSIS
             0000'CF     D5    035E            TSTL    W^MAC$GL_DIRFLG           ;WAS SYMBOL TABLE OUTPUT?
                  04     12    0362            BNEQ    10$                       ;IF NEQ YES
             0000'CF     D4    0364            CLRL    W^MAC$GL_LINE_CNT         ;NO--ALWAYS OUTPUT NEW PAGE
         00000000'GF     9F    0368     10$:   PUSHAB  G^MAC$AB_PSC_HD2A         ;CREATE THE HEADER MESSAGE BLOCK
         00000000'GF     9F    036E            PUSHAB  G^MAC$AB_PSC_HD2          ;DYNAMICALLY ON THE STACK.
                  00     DD    0374            PUSHL   #0                        ;-- 0 INDICATES A BLANK LINE --
         00000000'GF     9F    0376            PUSHAB  G^MAC$AB_PSC_HDRB         ;THIS WILL BE REMOVED LATER, ONCE
         00000000'GF     9F    037C            PUSHAB  G^MAC$AB_PSC_HDRA         ;THE NEW SUBTITLE AND SUBSECTION
         00000000'GF     9F    0382            PUSHAB  G^MAC$AB_PSC_HDR          ;HAVE BEEN ESTABLISHED.
                  00     DD    0388            PUSHL   #0                        ;-- 0 INDICATES A BLANK LINE --
                  07     DD    038A            PUSHL   #7                        ;NUMBER OF MESSAGE LINES.
                  5E     DD    038C            PUSHL   SP                        ;ADDRESS OF MESSAGE BLOCK.
                  05     DD    038E            PUSHL   #5                        ;STACK # OF LINES WE WILL USE
        6E   0000'CF     C0    0390            ADDL2   W^MAC$GL_PSC_MAX,(SP)     ;ADD TO GET TOTAL WE NEED
         00000000'EF     9F    0395            PUSHAB  L^MAC$AB_PSS_MSG          ;STACK SUBTITLE STRING
        0813'CF   03     FB    039B            CALLS   #3,W^NEW_SBT_CK_PAGE      ;DO NEW PAGE IF NEEDED AND OUTPUT HEADER
             50   8E     D0    03A0            MOVL    (SP)+, R0                 ;CLEAR THE MESSAGE BLOCK FROM THE STACK.
           5E   6E40     DE    03A3            MOVAL   (SP)[R0], SP              ;RESTORE THE STACK POINTER.
        56   0000'CF     9E    03A7            MOVAB   W^PSECT$MAIN,R6           ;POINT TO ABS PSECT
                053B     30    03AC            BSBW    MAC$PSECT_PRINT           ;PRINT ITS INFO
        56   0000'CF     9E    03AF            MOVAB   W^PSECT$BLANK,R6          ;POINT TO BLANK PSECT
       03 09 A6   07     E1    03B4            BBC     #SYM$V_REF, SYM$W_FLAG(R6),15$ ;HAS IT BEEN REFERENCED?
                052E     30    03B9            BSBW    MAC$PSECT_PRINT           ;YES, PRINT ITS INFO
        56   0000'CF     D0    03BC     15$:   MOVL    W^MAC$GL_PSC_LIST,R6      ;GET PSECT LIST POINTER
                  08     13    03C1            BEQL    30$                       ;IF EQL NO MORE
                0524     30    03C3     20$:   BSBW    MAC$PSECT_PRINT           ;PRINT THIS PSECT
             56   66     D0    03C6            MOVL    (R6),R6                   ;NEXT PSECT
                  F8     12    03C9            BNEQ    20$                       ;IF NEQ THERE IS A NEXT
             0000'CF     9F    03CB     30$:   PUSHAB  W^MAC$GQ_RNT_PSY          ;STACK TIMING BLOCK ADDRESS
        0000'CF   01     FB    03CF            CALLS   #1,W^MAC$TIMER_OFF        ;STOP TIMING PSECT SYNOPSIS
```

```
                                03D4                   .SBTTL  PRINT CROSS-REFERENCE IF REQUESTED
                                03D4
                                03D4       PRT_CROSS_REF:
               0000'CF    D5    03D4                   TSTL    W^MAC$GL_CRF_FLG            ;WAS THERE ANY CROSS REFERENCE?
                    6E    13    03D8                   BEQL    40$                         ;IF EQL NO--SKIP IT ALL
               0000'CF    9F    03DA                   PUSHAB  W^MAC$GQ_RNT_CRF            ;STACK TIMING BLOCK ADDRESS
        0000'CF     01    FB    03DE                   CALLS   #1,W^MAC$TIMER_ON           ;START TIMING CREF PHASE
                    59    D4    03E3                   CLRL    R9                          ;CLEAR INDEX INTO TABLES
    50   00000000'EF49    D0    03E5       10$:        MOVL    L^MAC$AL_CRF_TB5[R9],R0     ;ARE WE DONE?
                    50    13    03ED                   BEQL    30$                         ;IF EQL YES
    51   00000000'EF49    9A    03EF                   MOVZBL  L^MAC$AB_CRF_TB6[R9],R1     ;GET BIT # TO CHECK
     3E 0000'CF     51    E1    03F7                   BBC     R1,W^MAC$GL_CRF_FLG,20$     ;IF CLEAR TRY NEXT CREF AREA
                    60    D5    03FD                   TSTL    (R0)                        ;DID WE CREF ANYTHING?
                    3A    13    03FF                   BEQL    20$                         ;IF EQL NO--DO NEXT
              50    59    D0    0401                   MOVL    R9, R0                      ; Pass index.
                    51    D4    0404                   CLRL    R1                          ; Assume normal cref width.
    52   00000000'EF49    D0    0406                   MOVL    L^MAC$AL_CRF_TB3[R9],R2     ;GET CONTROL TABLE ADDRESS
    53   00000000'EF49    D0    040E                   MOVL    L^MAC$AL_CRF_TB7[R9],R3     ; Get column width flag
                    18    13    0416                   BEQL    15$                         ; If EQL not a variable field
 48 A2   00000000'EF49    D0    0418                   MOVL    L^MAC$AL_CRF_TB8[R9], -     ; Set KEY1FIELD entry
                                0421                           CRF$L_K1FMTB[(R2)           ; assuming narrow field
           0B 6B    53    E1    0421                   BBC     R3,(R11),15$                ; Branch if narrow column listing
                    51    D6    0425                   INCL    R1                          ; Set wide listing flag.
 48 A2   00000000'EF49    D0    0427                   MOVL    L^MAC$AL_CRF_TB9[R9], -     ; Set KEY1FIELD ENTRY
                                0430                           CRF$L_K1FMTB[(R2)           ; for wide field
                                0430       15$:
    53   00000000'EF49    9A    0430                   MOVZBL  L^MAC$AB_CRF_TB4[R9],R3     ;GET DEFS_REFS OR VALS_REFS
                  0354    30    0438                   BSBW    CREF_TREE_OUT               ;DO THIS CREF
                    59    D6    043B       20$:        INCL    R9                          ;NEXT CREF
                    A6    11    043D                   BRB     10$                         ;DO IT
               0000'CF    9F    043F       30$:        PUSHAB  W^MAC$GQ_RNT_CRF            ;STACK TIMING BLOCK ADDRESS
        0000'CF     01    FB    0443                   CALLS   #1,W^MAC$TIMER_OFF          ;STOP TIMING CROSS REFERENCE
               0000'CF    9F    0448       40$:        PUSHAB  W^MAC$GQ_RNT_TOT            ;STACK TIMING BLOCK ADDRESS
        0000'CF     01    FB    044C                   CALLS   #1,W^MAC$TIMER_OFF          ;STOP TIMING OF ASSEMBLER NOW
```

```
                                   0451   569          .SBTTL  OUTPUT CPU AND ELAPSED TIME STATISTICS
                                   0451   570
                                   0451   571 PRT_RUN_TIM:
               00000000'GF    9F   0451   572          PUSHAB  G^MAC$AB_RUN_HDRA         ;CREATE THE HEADER MESSAGE BLOCK
               00000000'GF    9F   0457   573          PUSHAB  G^MAC$AB_RUN_HDR          ;DYNAMICALLY ON THE STACK.
                        00    DD   045D   574          PUSHL   #0                        ;-- 0 INDICATES A BLANK LINE --
               00000000'GF    9F   045F   575          PUSHAB  G^MAC$AB_IND_HDRB         ;THIS WILL BE REMOVED LATER, ONCE
               00000000'GF    9F   0465   576          PUSHAB  G^MAC$AB_IND_HDRA         ;THE NEW SUBTITLE AND SUBSECTION
               00000000'GF    9F   046B   577          PUSHAB  G^MAC$AB_IND_HDR          ;HAVE BEEN ESTABLISHED.
                        00    DD   0471   578          PUSHL   #0                        ;-- 0 INDICATES A BLANK LINE --
                        07    DD   0473   579          PUSHL   #7                        ;NUMBER OF MESSAGE LINES.
                        5E    DD   0475   580          PUSHL   SP                        ;ADDRESS OF MESSAGE BLOCK.
                        08    DD   0477   581          PUSHL   #8                        ;# LINES NEEDED
               00000000'EF    9F   0479   582          PUSHAB  L^MAC$AB_RNT_MSG          ;STACK SUBTITLE STRING
              0813'CF    03   FB   047F   583          CALLS   #3,W^NEW_SBT_CK_PAGE      ;DO NEW PAGE IF NEEDED AND OUTPUT HEADER
                   50    8E   D0   0484   584          MOVL    (SP)+, R0                 ;CLEAR THE MESSAGE BLOCK FROM THE STACK.
                5E     6E40   DE   0487   585          MOVAL   (SP)[R0], SP              ;RESTORE THE STACK POINTER.
          59   00000000'EF    9E   048B   586          MOVAB   L^MAC$AW_IND_NAMS,R9      ;POINT TO TABLE OF NAMES
          58   00000000'EF    9E   0492   587          MOVAB   L^MAC$AW_RNT_PTRS,R8      ;POINT TO TABLE OF RUNTIME WORDS
          56   00000000'EF    9E   0499   588          MOVAB   L^MAC$AW_TIM_PTRS,R6      ;POINT TO TABLE OF ELAPSED QUADWORDS
                   50    89   3C   04A0   589 20$:     MOVZWL  (R9)+,R0                  ;GET ADDRESS OF STRING
                         3C   13   04A3   590          BEQL    50$                       ;IF EQL ALL DONE
                   51    88   3C   04A5   591          MOVZWL  (R8)+,R1                  ;GET ADDRESS OF CPU TIME QUADWORD
                         52   7C   04A8   592          CLRQ    R2                        ;CLEAR R2,R3
                         54   D4   04AA   593          CLRL    R4                        ;CLEAR R4 ALSO
 51     52    61   00000064 8F   7B   04AC   594          EDIV    #100,(R1),R2,R1           ;FRACTIONAL PART TO R1
         52     53     52    3C   7B   04B5   595          EDIV    #60,R2,R3,R2              ;SECONDS TO R2
         53     54     53    3C   7B   04BA   596          EDIV    #60,R3,R4,R3              ;MINUTES TO R3, HOURS IN R4
                   7E    66   3C   04BF   597 30$:     MOVZWL  (R6),-(SP)                ;GET ADDRESS OF ELAPSED TIME QUADWORD
                         51   DD   04C2   598          PUSHL   R1                        ;...
                         52   DD   04C4   599          PUSHL   R2
                         53   DD   04C6   600          PUSHL   R3
                         54   DD   04C8   601          PUSHL   R4
                   51    86   3C   04CA   602          MOVZWL  (R6)+,R1                  ;POINT TO ELAPSED TIME QUADWORD
                      08 A1   DD   04CD   603          PUSHL   8(R1)                     ;STACK THE # OF PAGE FAULTS
                         50   DD   04D0   604          PUSHL   R0                        ;STACK STRING ADDRESS
          50   00000000'EF    9E   04D2   605          MOVAB   L^MAC$AB_FAO_TIM,R0       ;POINT TO FAO CONTROL STRING
                     FB24'    30   04D9   606          BSBW    MAC$WRT_FAOUTS            ;FORMAT AND OUTPUT THE LINE
                   5E    1C   C0   04DC   607          ADDL2   #7*4,SP                   ;CLEAR THE ARG LIST FROM THE STACK
                         BF   11   04DF   608          BRB     20$
                                   04E1   609 50$:
```

```
                                04E1   611          .SBTTL  PRINT MEMORY USE STATISTICS
                                04E1   612
                                04E1   613 PRT_MEM_USE:
                    FB1C'   30  04E1   614          BSBW    MAC$WRT_BLNKLIN             ;SKIP A LINE
                                04E4   615          $ADJWSL_S PAGCNT=#0,WSETLM=W^MAC$GL_DIRFLG ;GET WS LIMIT
                0000'CF     DD  04F1   616          PUSHL   W^MAC$GL_DIRFLG             ;STACK WS LIMIT
      50    00000000'EF     9E  04F5   617          MOVAB   L^MAC$AB_WSL_FAO,R0         ;POINT TO FAO CONTROL STRING
                    FB01'   30  04FC   618          BSBW    MAC$WRT_FAOUTS              ;FORMAT AND OUTPUT
                5E    04    C0  04FF   619          ADDL2   #1*4,SP                     ;CLEAN THE STACK
      50    00000000'EF     D0  0502   620          MOVL    L^MAC$GL_INTPAGRQ,R0        ;GET # BYTES OF INT. BUFFER USED
  51  50    000001FF 8F     C1  0509   621          ADDL3   #^X1FF,R0,R1                ;ROUND UP TO PAGES
      7E    51    F7 8F     78  0511   622          ASHL    #-9,R1,-(SP)                ;CONVERT TO PAGES AND STACK
                      50    DD  0516   623          PUSHL   R0                          ;STACK # BYTES OF INT. BUFFER
      50    00000000'EF     9E  0518   624          MOVAB   L^MAC$AB_IFP_FAO,R0         ;POINT TO FAO STRING
                    FADE'   30  051F   625          BSBW    MAC$WRT_FAOUTS              ;FORMAT AND OUTPUT THE STRING
                5E    08    C0  0522   626          ADDL2   #2*4,SP                     ;CLEAR THE STACK
                0000'CF     DD  0525   627          PUSHL   W^MAC$GL_SYM_LOCL           ;STACK COUNT OF LOCAL SYMBOLS
                0000'CF     DD  0529   628          PUSHL   W^MAC$GL_SYM_NLOC           ;STACK COUNT OF NON-LOCAL SYMBOLS
                0000'CF     C5  052D   629          MULL3   W^MAC$GL_SYMPGREQ,-         ;COMPUTE # PAGES ALLOCATED
                50    0A        0531   630                  #STB$K_PG_MISS,R0           ;...
                      50    DD  0533   631          PUSHL   R0                          ;STACK # PAGES ALLOCATED
            00000000'EF     9F  0535   632          PUSHAB  L^MAC$AB_WERE_MG            ;ASSUME MANY PAGES ALLOCATED
          51    50    01    C3  053B   633          SUBL3   #1,R0,R1                    ;WAS THERE ONLY ONE PAGE?
                      07    12  053F   634          BNEQ    10$                         ;IF NEQ NO
      6E    00000000'EF     9E  0541   635          MOVAB   L^MAC$AB_WAS_MSG,(SP)       ;YES--USE "WAS" INSTEAD
      50    00000000'EF     9E  0548   636 10$:     MOVAB   L^MAC$AB_SYP_FAO,R0         ;POINT TO FAO CONTROL STRING
                    FAAE'   30  054F   637          BSBW    MAC$WRT_FAOUTS              ;FORMAT AND OUTPUT THE STRING
                5E    10    C0  0552   638          ADDL2   #4*4,SP                     ;CLEAR THE STACK
                0000'CF     DD  0555   639          PUSHL   W^MAC$GL_OBJ_RCNT           ;STACK OBJECT RECORD COUNT
          50    0000'CF     D0  0559   640          MOVL    W^MAC$GL_SRC_LCNT,R0        ;GET SOURCE RECORD COUNT
            00000000'EF     9F  055E   641          PUSHAB  L^MAC$AB_WERE_MG            ;ASSUME NOT 1 SOURCE LINE
          51    50    01    C3  0564   642          SUBL3   #1,R0,R1                    ;CHECK FOR 1 LINE
                      07    12  0568   643          BNEQ    20$                         ;IF NEQ NOT 1 LINE
      6E    00000000'EF     9E  056A   644          MOVAB   L^MAC$AB_WAS_MSG,(SP)       ;1 LINE--USE "WAS"
                      50    DD  0571   645 20$:     PUSHL   R0                          ;STACK SOURCE RECORD COUNT
      50    00000000'EF     9E  0573   646          MOVAB   L^MAC$AB_OBSR_FAO,R0        ;GET FAO CTL STRING ADDRESS
                    FA83'   30  057A   647          BSBW    MAC$WRT_FAOUTS              ;FORMAT AND OUTPUT THE STRING
                5E    0C    C0  057D   648          ADDL2   #3*4,SP                     ;CLEAR THE STACK
                0000'CF     DD  0580   649          PUSHL   W^MAC$GL_MCDEF              ;STACK # OF MACROS DEFINED
          50    0000'CF     D0  0584   650          MOVL    W^MAC$GL_MCPGRQ,R0          ;GET # PAGES REQUIRED TO DO SO
            00000000'EF     9F  0589   651          PUSHAB  L^MAC$AB_WERE_MG            ;ASSUME NOT 1 PAGE
          51    50    01    C3  058F   652          SUBL3   #1,R0,R1                    ;CHECK FOR ONE PAGE
                      07    12  0593   653          BNEQ    30$                         ;IF NEQ NOT 1 PAGE
      6E    00000000'EF     9E  0595   654          MOVAB   L^MAC$AB_WAS_MSG,(SP)       ;1 PAGE--USE "WAS"
                      50    DD  059C   655 30$:     PUSHL   R0                          ;STACK # PAGES
      50    00000000'EF     9E  059E   656          MOVAB   L^MAC$AB_MCD_FAO,R0         ;ADDRESS OF FAO CONTROL STRING
                    FA58'   30  05A5   657          BSBW    MAC$WRT_FAOUTS              ;FORMAT AND OUTPUT THE STRING
                5E    0C    C0  05A8   658          ADDL2   #3*4,SP                     ;CLEAN THE STACK
```

```
                              05AB   660            .SBTTL  PRINT MACRO LIBRARY USE STATISTICS
                              05AB   661
                              05AB   662 PRT_MLB_STATS:
          0000'CF       D5    05AB   663            TSTL    W^MAC$GL_MLB_CNT                ;WERE THERE ANY MLB'S?
               03       14    05AF   664            BGTR    10$                             ;IF GTR YES
             00B9       31    05B1   665            BRW     60$                             ;NO--SKIP ALL OF THIS
      00000000'GF       9F    05B4   666 10$:       PUSHAB  G^MAC$AB_MLB_HD2A               ;CREATE THE HEADER MESSAGE BLOCK
      00000000'GF       9F    05BA   667            PUSHAB  G^MAC$AB_MLB_HD2                ;DYNAMICALLY ON THE STACK.
               00       DD    05C0   668            PUSHL   #0                              ;-- 0 INDICATES A BLANK LINE --
      00000000'GF       9F    05C2   669            PUSHAB  G^MAC$AB_MLB_HDRB               ;THIS WILL BE REMOVED LATER, ONCE
      00000000'GF       9F    05C8   670            PUSHAB  G^MAC$AB_MLB_HDRA               ;THE NEW SUBTITLE AND SUBSECTION
      00000000'GF       9F    05CE   671            PUSHAB  G^MAC$AB_MLB_HDR                ;HAVE BEEN ESTABLISHED.
               00       DD    05D4   672            PUSHL   #0                              ;-- 0 INDICATES A BLANK LINE --
               07       DD    05D6   673            PUSHL   #7                              ;NUMBER OF MESSAGE LINES.
               5E       DD    05D8   674            PUSHL   SP                              ;ADDRESS OF MESSAGE BLOCK.
  7E    0000'CF  07     C1    05DA   675            ADDL3   #7,W^MAC$GL_MLB_CNT,-(SP)  ;FIGURE TOTAL LINES USED AND STACK
  50    0000'CF  01     C3    05E0   676            SUBL3   #1,W^MAC$GL_MLB_CNT,R0     ;WAS THERE ONLY 1 LIBRARY?
               02       15    05E6   677            BLEQ    20$                             ;IF LEQ YES
               6E       D6    05E8   678            INCL    (SP)                            ;NO--PRINT TOTALS LINE ALSO
      00000000'EF       9F    05EA   679 20$:       PUSHAB  L^MAC$AB_RNT_MSG                ;STACK LINE 2 HEADER MESSAGE
        0813'CF  03     FB    05F0   680            CALLS   #3,W^NEW_SBT_CK_PAGE            ;OUTPUT HEADER AND PAGE IF NECESSARY
             50  8E     D0    05F5   681            MOVL    (SP)+, R0                       ;CLEAR THE MESSAGE BLOCK FROM THE STACK.
          5E   6E40     DE    05F8   682            MOVAL   (SP)[R0], SP                    ;RESTORE THE STACK POINTER.
     56    0000'CF      D0    05FC   683            MOVL    W^MAC$GL_MLB_QUE,R6             ;POINT TO THE MLB QUEUE
                              0601   684 30$:
            08 A6       DD    0601   685            PUSHL   MLF$L_MCDEF(R6)                 ;STACK NUMBER OF MACROS DEFINED
            0C A6       9F    0604   686            PUSHAB  MLF$Q_FNAMDS(R6)                ; Stack file name descriptor address
  50    00000000'EF     9E    0607   687            MOVAB   L^MAC$AB_MLB_FAO,R0             ;GET FAO CONTROL STRING ADDRESS
            F9EF'       30    060E   688            BSBW    MAC$WRT_FAOUTS                  ;FORMAT AND OUTPUT THE STRING
          5E   08       C0    0611   689            ADDL2   #2*4,SP                         ; Clear the stack
          56   66       D0    0614   690 40$:       MOVL    (R6),R6                         ;LINK TO NEXT MLB
    00000000'8F  56     D1    0617   691            CMPL    R6,#MAC$GL_MLB_QUE              ;AT THE END OF THE QUEUE?
               E1       12    061E   692            BNEQ    30$                             ;IF NEQ NO
  50    0000'CF  01     C3    0620   693            SUBL3   #1,W^MAC$GL_MLB_CNT,R0          ;WAS THERE ONLY ONE MLB?
               17       15    0626   694            BLEQ    50$                             ;IF LEQ YES--SKIP TOTALS LINE
          0000'CF       DD    0628   695            PUSHL   W^MAC$GL_MLB_MDF                ;STACK TOTAL NUMBER OF MACROS DEFINED
      00000000'EF       9F    062C   696            PUSHAB  L^MAC$AB_MLB_TOT                ; Stack descriptor of Totals string
  50    00000000'EF     9E    0632   697            MOVAB   L^MAC$AB_MLB_FAO,R0             ;POINT TO FAO CONTROL STRING
            F9C4'       30    0639   698            BSBW    MAC$WRT_FAOUTS                  ;FORMAT AND OUTPUT THE STRING
          5E   08       C0    063C   699            ADDL2   #2*4,SP                         ; Clear the stack
            F9BE'       30    063F   700 50$:       BSBW    MAC$WRT_BLNKLIN                 ;SKIP A LINE
          0000'CF       DD    0642   701            PUSHL   W^MAC$GL_MLB_MDF                ;STACK # MACROS DEFINED
     50   0000'CF       D0    0646   702            MOVL    W^MAC$GL_MLB_GET,R0             ;GET # GETS REQUIRED
      00000000'EF       9F    064B   703            PUSHAB  L^MAC$AB_WERE_MG                ;ASSUME MANY GETS NEEDED
     51    50    01     C3    0651   704            SUBL3   #1,R0,R1                        ;SEE IF ONLY 1 GET
               07       12    0655   705            BNEQ    55$                             ;IF NEQ NOT 1 GET
  6E    00000000'EF     9E    0657   706            MOVAB   L^MAC$AB_WAS_MSG,(SP)           ;1 GET--SAY 'WAS'
               50       DD    065E   707 55$:       PUSHL   R0                              ;STACK # GETS REQUIRED
  50    00000000'EF     9E    0660   708            MOVAB   L^MAC$AB_MLB_SUM,R0             ;POINT TO FAO CONTROL STRING
            F996'       30    0667   709            BSBW    MAC$WRT_FAOUTS                  ;FORMAT AND OUTPUT STRING
          5E   0C       C0    066A   710            ADDL2   #3*4,SP                         ;CLEAR STACK
                              066D   711 60$:
                                  066D          .SBTTL  PRINT ERROR SUMMARY
                                  066D
                                  066D  PRT_ERR_SUM:
                   F990'    30    066D          BSBW    MAC$WRT_BLNKLIN          ;SKIP A LINE
        5A   00000000'EF    D0    0670          MOVL    L^MAC$GL_ERRCT,R10       ;GET THE ERROR TOTAL
           59      0000'CF  D0    0677          MOVL    W^MAC$GL_WARNCT,R9       ;AND THE WARNINGS
           58      0000'CF  D0    067C          MOVL    W^MAC$GL_INFOCNT,R8      ; and the information messages
           50   59      5A  C1    0681          ADDL3   R10,R9,R0                ;WERE THERE ANY ERRORS/WARNINGS?
                50      58  C0    0685          ADDL2   R8,R0                    ; or information messages
                        0D  12    0688          BNEQ    10$                      ;IF NEQ YES
        50   00000000'EF    9E    068A          MOVAB   L^MAC$AB_NOERRM,R0       ;NO--GET NO ERRORS MESSAGE
                   F96C'    30    0691          BSBW    MAC$WRT_FAOUTS           ;FORMAT AND OUTPUT IT
                    00BB    31    0694          BRW     PRT_ERR_SUM_END          ;ALL DONE
                                  0697  ;
                                  0697  ; THERE WERE ERRORS/WARNINGS. PRINT THAT.
                                  0697  ;
                                  0697  10$:
                        58  DD    0697          PUSHL   R8                       ; Stack information messages count
                        59  DD    0699          PUSHL   R9                       ; warning count
                        5A  DD    069B          PUSHL   R10                      ;AND ERROR COUNT
        50   00000000'EF    9E    069D          MOVAB   L^MAC$AB_ERRM,R0         ;GET THE FAO STRING
                   F959'    30    06A4          BSBW    MAC$FAOUTS               ;FORMAT THE LINE
                5E      0C  C0    06A7          ADDL2   #3*4,SP                  ; Pop three longwords from stack
                   F953'    30    06AA          BSBW    MAC$TERM_BLANK           ;WRITE ONE BLANK LINE TO TERMINAL
           50      0000'CF  D0    06AD          MOVL    W^MAC$GL_LINELN,R0       ;GET THE LINE LENGTH
                   F94B'    30    06B2          BSBW    MAC$WRITE_TERM           ;WRITE LINE TO TERMINAL
                   F948'    30    06B5          BSBW    MAC$WRTLST               ;AND THE LISTING FILE
        5A   00000000'EF    D0    06B8          MOVL    L^MAC$GL_ERR_LIST,R10    ;GET PTR TO FIRST ERROR BLOCK
             00000004'FF    D4    06BF          CLRL    @L^MAC$GL_ERR_LIST+4     ;CLEAR LINK WORD IN LAST BLOCK
             59      10 AA  9E    06C5          MOVAB   16(R10),R9               ;POINT TO FIRST PAGE/LINE
  58   0000007E 8F   0C AA  C3    06C9          SUBL3   12(R10),#<<1024-16>/8>,R8 ;FIGURE PAGE/LINES IN THIS BLOCK
                                  06D2  ;
                                  06D2  ; BACK HERE FOR EACH NEW LINE OF LINE/PAGE NUMBERS
                                  06D2  ;
                        05  DD    06D2  20$:    PUSHL   #5                       ;FIVE LINE/PAGES PER LISTING LINE
    0004'CF  00000000'EF    9E    06D4          MOVAB   L^MAC$AB_LST_END,W^MAC$G_ERRBFDES+4 ;SET BUFFER DESCRIPTOR
                                  06DD  ;
                                  06DD  ; HERE FOR EACH NEW LINE/PAGE ITEM ON A PAGE
                                  06DD  ;
                        89  DD    06DD  30$:    PUSHL   (R9)+                    ;STACK THE PAGE NUMBER
                50      01  D0    06DF          MOVL    #1,R0                    ; Assume insert number exists
                7E      89  3C    06E2          MOVZWL  (R9)+,-(SP)              ; Stack insert number
                        06  12    06E5          BNEQ    35$                      ; If NEQ insert number did exists
                        50  D4    06E7          CLRL    R0                       ; Set no insert number
             6E      04 AE  D0    06E9          MOVL    4(SP),(SP)               ; and move up line number
                                  06ED  35$:
                        50  DD    06ED          PUSHL   R0                       ; Stack number of insert numbers (0 or 1)
                        50  DD    06EF          PUSHL   R0                       ; Stack number of '.'s (0 or 1)
                7E      89  3C    06F1          MOVZWL  (R9)+,-(SP)              ; Stack line number
                50      5E  D0    06F4          MOVL    SP,R0                    ;COPY STACK ADDRESS
                                  06F7          $FAOL_S CTRSTR=L^MAC$AB_LPG_FMT,- ;FORMAT THE LINE/PAGE
                                  06F7                  OUTBUF=L^MAC$G_ERRBFDES,-
                                  06F7                  OUTLEN=L^MAC$GL_LINELN,-
                                  06F7                  PRMLST=(R0)
           0004'CF      0E  C0    0712          ADDL2   #14,W^MAC$G_ERRBFDES+4   ;UPDATE POINTER FOR NEXT FAO (12 BYTES/LINE/
                5E      14  C0    0717          ADDL2   #5*4,SP                  ;DELETE FIVE LONGWORDS FROM STACK
                     12 58  F5    071A          SOBGTR  R8,40$                   ;BRANCH IF NOT DONE THIS BLOCK
             5A    6A   D0  071D   770            MOVL    (R10),R10                 ;YES--LINKE TO NEXT
                   10   13  0720   771            BEQL    50$                       ;IF EQL ALL DONE
              59   10 AA   9E  0722   772            MOVAB   16(R10),R9                ;POINT TO FIRST LINE/PAGE
   58   0000007E 8F   0C AA   C3  0726   773            SUBL3   12(R10),#<<1024-16>/8>,R8 ;FIGURE LINE/PAGES IN THIS BLOCK
                AB 6E   F5  072F   774 40$:       SOBGTR  (SP),30$                  ;LOOP FOR 5 LINE/PAGES
                   8E   D5  0732   775 50$:       TSTL    (SP)+                     ;CLEAR STACK
 0004'CF   00000000'8F   C3  0734   776            SUBL3   #MAC$AB_LINEBF,W^MAC$G_ERRBFDES+4,- ;FIGURE LINE LENGTH
                   50       073D   777                                    R0
      00000000'EF   50   D0  073E   778            MOVL    R0,L^MAC$GL_LINELN        ;SAVE LINE LENGTH
                F8B8'   30  0745   779            BSBW    MAC$WRITE_TERM            ;WRITE LINE TO TERMINAL
                F8B5'   30  0748   780            BSBW    MAC$WRTLST                ;WRITE THE LINE
                   5A   D5  074B   781            TSTL    R10                       ;ANY MORE TO DO?
                   83   12  074D   782            BNEQ    20$                       ;IF NEQ YES
                F8AE'   30  074F   783            BSBW    MAC$TERM_BLANK            ;BLANK LINE TO TERMINAL
                            0752   784 PRT_ERR_SUM_END:
                                  0752   786           .SBTTL  PRINT COMMAND LINE
                                  0752   787
                                  0752   788 PRT_CMD_LIN:
                       F8AB'   30 0752   789           BSBW    MAC$WRT_BLNKLIN          ;SKIP A LINE
                  00000000'EF  DD 0755   790           PUSHL   L^MAC$GL_CMDLIN          ;STACK COMMAND LINE ADDRESS
                  00000000'EF  DD 075B   791           PUSHL   L^MAC$GL_CMDLEN          ;AND ITS LENGTH
            50    00000000'EF  9E 0761   792           MOVAB   L^MAC$AB_CMD_FAO,R0      ;POINT TO FAO CONTROL STRING
                       F895'   30 0768   793           BSBW    MAC$FAOUTS               ;FORMAT THE STRING
50   00000000'EF  00000000'EF  C1 076B   794           ADDL3   L^MAC$GL_ERRCT,L^MAC$GL_WARNCT,R0 ;ERRORS OR WARNINGS?
                  50    0000'CF  C0 0777   795           ADDL2   W^MAC$GL_INFOCNT,R0      ; or information messages?
                         03    13 077C   796           BEQL    10$                      ;IF EQL NO
                       F87F'   30 077E   797           BSBW    MAC$WRITE_TERM           ;YES--WRITE TO TERMINAL ALSO
                       F87C'   30 0781   798 10$:      BSBW    MAC$WRTLST               ;WRITE TO LISTING FILE
                    5E   08    C0 0784   799           ADDL2   #2*4,SP                  ;CLEAR 2 LW FROM STACK
                                  0787   800 FIN_ASM_EXIT:
                       5A  8ED0 0787   801           POPL    R10                      ;RESTORE R10
                   0380 8F     BA 078A   802           POPR    #^M<R7,R8,R9>            ;RESTORE REGISTERS
                               05 078E   803           RSB                              ;**EXIT FROM MAC$FINISH_ASM
```

```
                              078F           .SBTTL  OUTPUT ONE PHASE OF CROSS REFERENCE
                              078F
                              078F  ;++
                              078F  ; FUNCTIONAL DESCRIPTION:
                              078F  ;
                              078F  ;       THIS ROUTINE OUTPUTS ONE PHASE OF THE CROSS REFERENCE.
                              078F  ;
                              078F  ; INPUTS:
                              078F  ;
                              078F  ;       R0      CROSS REFERENCE TYPE INDEX
                              078F  ;       R1      0 => NORMAL WIDTH LISTING, 1 => WIDE LISTING FORMAT
                              078F  ;       R2      POINTER TO CREF CONTROL TABLE
                              078F  ;       R3      CRF$K_VALS_REFS OR CRF$K_DEFS_REFS
                              078F  ;
                              078F  ;--
                              078F
                              078F  CREF_TREE_OUT:
     00000000'EF      D4      078F          CLRL    L^MAC$GL_LINE_CNT         ;ALWAYS FORCE NEW PAGE FOR CREF
              53      DD      0795          PUSHL   R3                        ;SAVE REFS OR DEFS_REFS FLAG
              52      DD      0797          PUSHL   R2                        ;SAVE CREF CONTROL BLOCK ADDRESS
           10 51      E9      0799          BLBC    R1, 10$                   ;CHECK THE LISTING CONTROL INDICATOR.
   00000000'GF40      DD      079C          PUSHL   G^MAC$AL_CRF_TB2WA[R0]    ;WIDE LISTING FORMAT HAS BEEN CHOSEN.
   00000000'GF40      DD      07A3          PUSHL   G^MAC$AL_CRF_TB2W [R0]    ;STACK THE SECONDARY HEADER LINES.
              0E      11      07AA          BRB     20$                       ;JOIN COMMON CODE FOR SECTION HEADING.
   00000000'GF40      DD      07AC  10$:    PUSHL   G^MAC$AL_CRF_TB2A [R0]    ;NORMAL WIDTH LISTING HAS BEEN CHOSEN.
   00000000'GF40      DD      07B3          PUSHL   G^MAC$AL_CRF_TB2  [R0]    ;STACK THE SECONDARY HEADER LINES.
              00      DD      07BA  20$:    PUSHL   #0                        ;-- 0 INDICATES A BLANK LINE --
   00000000'GF40      DD      07BC          PUSHL   G^MAC$AL_CRF_TB1B [R0]    ;STACK THE SECTION HEADER LINES...
   00000000'GF40      DD      07C3          PUSHL   G^MAC$AL_CRF_TB1A [R0]    ; (THE ONES THAT FORM THE BOXED IN
   00000000'GF40      DD      07CA          PUSHL   G^MAC$AL_CRF_TB1  [R0]    ;  SECTION HEADERS).
              00      DD      07D1          PUSHL   #0                        ;-- 0 INDICATES A BLANK LINE --
              07      DD      07D3          PUSHL   #7                        ;NUMBER OF HEADER LINES IN THIS BLOCK.
              5E      DD      07D5          PUSHL   SP                        ;HEADER MESSAGE BLOCK ADDRESS.
              09      DD      07D7          PUSHL   #5+4                      ;STACK # LINES ACTUALLY NEEDED
     00000000'GF      9F      07D9          PUSHAB  G^MAC$AB_CRF_MSG          ;STACK LINE 2 HEADER MESSAGE
     0B13'CF  03      FB      07DF          CALLS   #3,W^NEW_SBT_CK_PAGE      ;DO HEADER THING
           50 8E      D0      07E4          MOVL    (SP)+, R0                 ;CLEAR THE MESSAGE BLOCK FROM THE STACK.
     5E    6E40       DE      07E7          MOVAL   (SP)[R0], SP              ;RESTORE THE STACK POINTER.
           50 8ED0            07EB          POPL    R0                        ;GET CREF CONTROL BLOCK ADDRESS
           51 8ED0            07EE          POPL    R1                        ;GET THE REFS OR DEFS_REFS FLAG
     00000000'8F      DD      07F1          PUSHL   #CRF$K_DELETE             ;DELETE TREE FLAG FOR CREF
              51      DD      07F7          PUSHL   R1                        ;SET REFS OR DEFS_REFS FLAG ON STACK
         0000'CF      DD      07F9          PUSHL   W^MAC$GL_LN_PAGE          ;# LINES ON SUBSEQUENT PAGES
     00000000'EF      DD      07FD          PUSHL   L^MAC$GL_LINE_CNT         ;# LINES ON FIRST PAGE
     00000084 8F      DD      0803          PUSHL   #132                      ;WIDTH OF A LINE
              50      DD      0809          PUSHL   R0                        ;CREF CONTROL BLOCK ADDRESS
 00000000'GF  06      FB      080B          CALLS   #6,G^CRF$OUT              ;OUTPUT CROSS REFERENCE
                      05      0812          RSB
```

```
                                  0813   854        .SBTTL  SET UP FOR NEW SUBTITLE AND CHECK NEW PAGE
                                  0813   855
                                  0813   856 ;++
                                  0813   857 ; FUNCTIONAL DESCRIPTION:
                                  0813   858 ;
                                  0813   859 ;      THIS ROUTINE PUTS A NEW LINE IN THE SUBTITLE BUFFER FOR
                                  0813   860 ;      PSECT SYNOPSIS AND THE RUN STATISTICS.
                                  0813   861 ;
                                  0813   862 ; INPUTS:
                                  0813   863 ;
                                  0813   864 ;      4(AP)   ADDRESS OF STRING TO PUT IN TITLE BUFFER (COUNT,TEXT)
                                  0813   865 ;      8(AP)   # LINES NEEDED LEFT ON PAGE
                                  0813   866 ;      12(AP)  ADDRESS OF HEADER MESSAGE BLOCK
                                  0813   867 ;
                                  0813   868 ;      NOTE:   THE FORMAT OF THE HEADER MESSAGE BLOCK IS THE SAME AS
                                  0813   869 ;              A STANDARD VAX/VMS ARGUMENT LIST, WHOSE ENTRIES ARE THE
                                  0813   870 ;              ADDRESSES OF FAO CONTROL STRING DESCRIPTORS.
                                  0813   871 ;--
                                  0813   872
                                  0813   873
                            0004  0813   874        .ENTRY NEW_SBT_CK_PAGE, ^M<R2>  ;ENTRY POINT
              50    04 AC    D0   0815   875        MOVL    4(AP),R0                ;GET SUBTITLE STRING ADDRESS
                 51    80    9A   0819   876        MOVZBL  (R0)+,R1                ;GET ITS LENGTH
           20    60    51    2C   081C   877        MOVC5   R1,(R0),#^A/ /,-        ;COPY OVER SUBTITLE AND IDENT
     00000000'EF     0048 8F      0820   878                #LST$K_TITLE_SIZ+SYM$K_MAXLEN+1, -
                                  0828   879                        L^MAC$AB_SBT_IDNT
  50   00000000'EF    08 AC  C3   0828   880        SUBL3   8(AP),L^MAC$GL_LINE_CNT,R0 ;SEE IF ROOM LEFT ON PAGE
                       03    14   0831   881        BGTR    10$                     ;IF GTR YES
                    F7CA'    30   0833   882        BSBW    MAC$LST_PAG_HDR         ;NO--MAKE NEW PAGE NOW
              52    0C AC    D0   0836   883 10$:   MOVL    12(AP),R2               ;GET ADDRESS OF HEADER MESSAGE BLOCK
                 51    82    D0   083A   884        MOVL    (R2)+, R1               ;GET THE NUMBER OF HEADER MESSAGE LINES
                       14    13   083D   885        BEQL    50$                     ;ANY PROVIDED? RETURN IF NOT.
              50       82    D0   083F   886 20$:   MOVL    (R2)+, R0               ;GET THE NEXT HEADER LINE
                       06    BB   0842   887        PUSHR   #^M<R1,R2>              ;PRESERVE REGISTERS ACROSS CALLS
                       05    13   0844   888        BEQL    30$                     ;BLANK LINE?
                    F7B7'    30   0846   889        BSBW    MAC$WRT_FAOUTS          ;NO, FORMAT AND OUTPUT THE STRING
                       03    11   0849   890        BRB     40$                     ;CHECK FOR MORE HEADER LINES...
                    F7B2'    30   084B   891 30$:   BSBW    MAC$WRT_BLNKLIN         ;SKIP A LINE.
                       06    BA   084E   892 40$:   POPR    #^M<R1,R2>              ;RESTORE REGISTERS.
                    EC 51    F5   0850   893        SOBGTR  R1, 20$                 ;ANY MORE HEADER LINES?
                             04   0853   894 50$:   RET                             ;ALL DONE
```

```
                          0854          .SBTTL  OUTPUT SYMBOL NAME AND VALUE FOR DEBUG
                          0854
                          0854  ;++
                          0854  ; FUNCTIONAL DESCRIPTION:
                          0854  ;
                          0854  ;       THIS ROUTINE OUTPUTS THE NAME OF A SYMBOL FOR THE DEBUGGER.
                          0854  ;
                          0854  ; INPUTS:
                          0854  ;
                          0854  ;       R6      POINTS TO SYMBOL BLOCK
                          0854  ;
                          0854  ;--
                          0854
                          0854  MAC$DBG_NAM_OUT:
     54    04 A6     9A   0854          MOVZBL  SYM$B_NAME(R6),R4         ; Get offset to name/count (which is
                          0858                                            ; also total size of count and name
   55    56    54    C3   0858          SUBL3   R4,R6,R5                  ; Set pointer to count/name
        50    85     90   085C  10$:    MOVB    (R5)+,R0                  ;GET CHARACTER OF NAME
           F79E'     30   085F          BSBW    MAC$STOIM
          F7 54      F5   0862          SOBGTR  R4,10$                    ;LOOP FOR WHOLE NAME
                     05   0865          RSB
                          0866
                          0866  ;++
                          0866  ; FUNCTIONAL DESCRIPTION:
                          0866  ;
                          0866  ; THIS ROUTINE OUTPUTS THE VALUE OF A SYMBOL FOR THE DEBUGGER.
                          0866  ;
                          0866  ; INPUTS:
                          0866  ;
                          0866  ;       R6      POINTS TO SYMBOL BLOCK
                          0866  ;
                          0866  ;--
                          0866
                          0866  MAC$DBG_VAL_OUT:
                          0866          $OBJ_CHKBYT #TIR$C_STA_PL         ;STACK PSECT PLUS LONGOWORD
     7E    0C A6     9A   086C          MOVZBL  SYM$B_SEG(R6), -(SP)      ;GET THE SEGMENT NUMBER.
              0A     13   0870          BEQL    10$                       ;DON'T FIDDLE WITH THE ABS PSECT!
              07     E0   0872          BBS     #SYM$V_REF,-              ;HAS THE BLANK PSECT BEEN REF'D?
 02 00000009'GF           0874                  G^PSECT$BLANK+SYM$W_FLAG,10$ ;IF NOT, IT'LL BE REMOVED, SO DECR
              6E     D7   087A          DECL    (SP)                      ;THE SEG# TO PRESERVE PSECT ALIGNMENT.
                          087C  10$:    $OBJ_OUTBYT (SP)                  ;EMIT THE SYMBOL'S SEGMENT #
              8E     D5   0882          TSTL    (SP)+                     ;CLEAN UP THE STACK
     55    05 A6     9E   0884          MOVAB   SYM$L_VAL(R6),R5          ;POINT TO VALUE
           F775'     30   0888          BSBW    MAC$OUT_LW                ;OUTPUT VALUE
                          088B          $OBJ_CHKBYT #TIR$C_STO_LW         ;STORE LONGWORD
                     05   0891          RSB
```

```
                         0892          .SBTTL  OUTPUT PSECT RECORDS TO DEBUG
                         0892
                         0892  ;++
                         0892  ; FUNCTIONAL DESCRIPTION:
                         0892  ;
                         0892  ;       THIS ROUTINE OUTPUTS PSECT INFORMATION TO THE DEBUGGER
                         0892  ;
                         0892  ; INPUTS:
                         0892  ;
                         0892  ;       R6      POINTS TO SYMBOL BLOCK FOR PSECT
                         0892  ;
                         0892  ;--
                         0892
                         0892  MAC$DBG_PSECT:
       50   04 A6    9A  0892          MOVZBL  SYM$B_NAME(R6),R0       ; Get offset to symbol count/name
    50   56   50    C3   0896          SUBL3   R0,R6,R0                ; Form address of count/name
    50   0B   60    81   089A          ADDB3   (R0),#DBG$K_PSECT_LEN-1,R0 ; Figure size
            F75F'   30   089E          BSBW    MAC$STOIM               ;STORE IT
       50   B8 8F   9A   08A1          MOVZBL  #DBG$C_PSECT,R0         ;PSECT TYPE
            F758'   30   08A5          BSBW    MAC$STOIM
               50   D4   08A8          CLRL    R0                      ;MBZ BYTE
            F753'   30   08AA          BSBW    MAC$STOIM
                         08AD          $OBJ_CHKBYT #TIR$C_STA_PB       ;STACK PSECT BASE PLUS BYTE OFFSET
       7E   0C A6   9A   08B3          MOVZBL  SYM$B_SEG(R6), -(SP)    ;GET THE SEGMENT NUMBER.
               0A   13   08B7          BEQL    5$                      ;DON'T FIDDLE WITH THE ABS PSECT!
               07   E0   08B9          BBS     #SYM$V_REF,-            ;HAS THE BLANK PSECT BEEN REF'D?
  02 00000009'GF         08BB                  G^PSECT$BLANK+SYM$W_FLAG,5$ ;IF NOT, IT'LL BE REMOVED, SO DECR
               6E   D7   08C1          DECL    (SP)                    ;THE SEG# TO PRESERVE PSECT ALIGNMENT.
                         08C3  5$:     $OBJ_OUTBYT (SP)                ;EMIT THE SYMBOL'S SEGMENT #
               8E   D5   08C9          TSTL    (SP)+                   ;AND CLEAN UP THE STACK.
               50   D4   08CB          CLRL    R0                      ;USE OFFSET OF 0
            F730'   30   08CD          BSBW    MAC$OUTOBJ
                         08D0          $OBJ_CHKBYT #TIR$C_STO_PIDR     ;STORE POSITION INDEPENDENT DATA REFERENCE.
            FF7B    30   08D6          BSBW    MAC$DBG_NAM_OUT         ;SEND PSECT NAME
       55   05 A6   9E   08D9          MOVAB   SYM$L_VAL(R6),R5        ;POINT TO LENGTH
          54   04   9A   08DD          MOVZBL  #4,R4                   ;LOOP COUNT
          50   85   90   08E0  10$:    MOVB    (R5)+,R0                ;GET BYTE OF VALUE
            F71A'   30   08E3          BSBW    MAC$STOIM               ;
            F7 54   F5   08E6          SOBGTR  R4,10$
                    05   08E9          RSB
```

```
                                          08EA          .SBTTL  PRINT PSECT INFORMATION FOR 1 PSECT
                                          08EA
                                          08EA  ;++
                                          08EA  ; FUNCTIONAL DESCRIPTION:
                                          08EA  ;
                                          08EA  ;       THIS ROUTINE PRINTS THE PSECT SYNOPSIS FOR ONE PSECT.
                                          08EA  ;
                                          08EA  ; INPUTS:
                                          08EA  ;
                                          08EA  ;       R6      PSECT BLOCK ADDRESS
                                          08EA  ;
                                          08EA  ;--
                                          08EA
                                          08EA  MAC$PSECT_PRINT:
                              01C6   30   08EA          BSBW    MAC$LIST_INIT                 ;INIT LISTING BUFFER
                        50   0C A6   9A   08ED          MOVZBL  PSC$B_SEG(R6),R0              ;GET SEGMENT NUMBER
                                0A   13   08F1          BEQL    2$                            ;DON'T FIDDLE WITH THE ABS PSECT.
                                07   E0   08F3          BBS     #SYM$V_REF,-                  ;HAS THE BLANK PSECT BEEN REF'D?
                    02 00000009'GF        08F5                  G^PSECT$BLANK+SYM$W_FLAG,2$ ;IF NOT, IT'LL BE REMOVED, SO DECR
                                50   D7   08FB          DECL    R0                            ;THE SEG# TO PRESERVE PSECT ALIGNMENT.
                                50   DD   08FD  2$:     PUSHL   R0                            ;STACK IT
                                50   DD   08FF          PUSHL   R0                            ;TWICE FOR FAO
                             05 A6   DD   0901          PUSHL   PSC$L_MAXLGTH(R6)             ;STACK THE MAX LENGTH
                                6E   DD   0904          PUSHL   (SP)                          ;COPY IT (HEX AND DEC OUTPUT)
                                03   14   0906          BGTR    5$                            ;BRANCH IF POSITIVE LENGTH
                             04 AE   D4   0908          CLRL    4(SP)                         ;ELSE PRINT DECIMAL AS 0
                        50   04 A6   9A   090B  5$:     MOVZBL  PSC$B_NAME(R6),R0             ; Get offset to symbol count/name
                      7E   56   50   C3   090F          SUBL3   R0,R6,-(SP)                   ; and put its address on stack
                 50    00000000'EF   9E   0913          MOVAB   L^MAC$AB_PSS_FAO,R0           ;POINT TO FAO CONTROL STRING
                             F6E3'   30   091A          BSBW    MAC$FAOUTS                    ;FORMAT STRING
                           5E   14   C0   091D          ADDL2   #5*4,SP                       ;CLEAR THE STACK
  50    00000000'EF    00000000'8F   C1   0920          ADDL3   #MAC$K_LIST_SIZE,L^MAC$GL_LINELN,R0 ;FIGURE LENGTH OF LINE SO FAR
                       00000000'E0   9E   092C          MOVAB   L^MAC$AB_LST_END(R0),-        ;INIT LISTING POINTER
                       00000000'EF        0932                          L^MAC$GL_LIST_PTR
                        55   0D A6   3C   0937          MOVZWL  PSC$W_OPTIONS(R6),R5          ;GET PSECT OPTIONS
                           54   01   D0   093B          MOVL    #1,R4                         ;FIRST BIT
                                0A   DD   093E          PUSHL   #PSC$K_NO_OPTNS               ; Loop once for each option
                                          0940  ;
                                          0940  ; LOOP, PRINTING PSECT OPTIONS
                                          0940  ;
                              003C   30   0940  10$:    BSBW    PSECT_OPT_MATCH               ;PRINT THIS OPTION
                           54   54   C0   0943          ADDL2   R4,R4                         ;NEXT BIT
                             F7 6E   F5   0946          SOBGTR  (SP),10$                      ;LOOP FOR ALL
                                8E   D5   0949          TSTL    (SP)+                         ;CLEAN STACK
                        54   0D A6   3C   094B          MOVZWL  PSC$W_OPTIONS(R6),R4          ;COPY OPTIONS AGAIN
                  54   000003FF 8F   CA   094F          BICL2   #^XPSC$M_ALLOPTNS,R4          ; Trim all but alignment
                      54   4000 8F   A8   0956          BISW2   #PSC$M_ALIGNFLG,R4            ;SET THE ALIGNMENT FLAG BIT
                           55   54   D0   095B          MOVL    R4,R5                         ;POSITIVE SENSE
                              001E   30   095E          BSBW    PSECT_OPT_MATCH               ;PRINT ALIGNMENT
                             08 50   E8   0961          BLBS    R0,20$                        ;BRANCH IF FOUND
                   50   54   F6 8F   78   0964          ASHL    #-PSC$V_ALIGNMENT,R4,R0       ;GET ALIGNMENT
                             F694'   30   0969          BSBW    MAC$DEC_OUT_L2R               ;OUTPUT ALIGNMENT
        00000000'EF    00000000'8F   C3   096C  20$:    SUBL3   #MAC$AB_LINEBF,L^MAC$GL_LIST_PTR,- ;FIGURE LINE LENGTH
                       00000000'EF        0977                          L^MAC$GL_LINELN
                             F681'   31   097C          BRW     MAC$WRTLST                    ;WRITE LINE TO LISTING AND RETURN
```

```
                          097F        .SBTTL  MATCH PSECT OPTIONS FOR PRINTING
                          097F
                          097F  ;++
                          097F  ; FUNCTIONAL DESCRIPTION:
                          097F  ;
                          097F  ;       THIS ROUTINE PRINTS EITHER 'XXX' OR 'NOXXX' FOR A GIVEN
                          097F  ;       PSECT OPTION.
                          097F  ;
                          097F  ; INPUTS:
                          097F  ;
                          097F  ;       R4      BIT VALUE
                          097F  ;       R5      OPTIONS
                          097F  ;
                          097F  ;--
                          097F
                          097F  PSECT_OPT_MATCH:
           50    54   B0  097F          MOVW    R4,R0                   ;COPY BITS
                 08   13  0982          BEQL    10$                     ;IF EQL NOT COMPLEMENTED
           55    50   B3  0984          BITW    R0,R5                   ;NEGATIVE POLARITY?
                 03   12  0987          BNEQ    10$                     ;IF NEQ NO
           50    50   B2  0989          MCOMW   R0,R0                   ;YES--GET COMPLEMENTED VALUE
53    00000000'EF     9E  098C  10$:    MOVAB   L^PSC$G_OPTIONS,R3      ;POINT TO PSECT OPTIONS
        07 A3    50   B1  0993  20$:    CMPW    R0,SYM$[_VAL+2(R3)      ;IS THIS IT?
                 08   13  0997          BEQL    30$                     ;IF EQL YES
           53    63   D0  0999          MOVL    SYM$L_LINK(R3),R3       ; No--link to next
                 F5   12  099C          BNEQ    20$                     ;IF NEQ GO ON
                 50   D4  099E          CLRL    R0                      ;RETURN 0 FOR NOT FOUND
                 05       09A0          RSB
        50    04 A3   9A  09A1  30$:    MOVZBL  SYM$B_NAME(R3),R0       ; Get offset to count/name
      50    53   50   C3  09A5          SUBL3   R0,R3,R0                ; and form its address
            51   80   9A  09A9          MOVZBL  (R0)+,R1                ;COPY THE SIZE
52    00000000'EF     D0  09AC          MOVL    L^MAC$GL_LIST_PTR,R2    ;GET LISTING POINTER
                 30   BB  09B3          PUSHR   #^M<R4,R5>              ;SAVE OPTION PARAMETERS
           03    51   91  09B5          CMPB    R1,#3                   ;IS IT LESS THAN 3-CHAR NAME?
                 02   14  09B8          BGTR    40$                     ;IF GTR NO
                 82   B5  09BA          TSTW    (R2)+                   ;YES--BUMP POINTER BY TWO
      62    60   51   28  09BC  40$:    MOVC3   R1,(R0),(R2)            ;COPY INTO BUFFER
                 30   BA  09C0          POPR    #^M<R4,R5>              ;RESTORE OPTION PARAMS
00000000'EF      06   C0  09C2          ADDL2   #6,L^MAC$GL_LIST_PTR    ;TAB TO NEXT OPTION
           50    01   9A  09C9          MOVZBL  #1,R0                   ;RETURN FOUND
                 05       09CC          RSB
```

```
                           09CD          .SBTTL  PRINT SYMBOL INFORMATION FOR ONE SYMBOL
                           09CD
                           09CD  ;++
                           09CD  ; FUNCTIONAL DESCRIPTION:
                           09CD  ;
                           09CD  ;         THIS ROUTINE PRINTS THE SYMBOL INFORMATION FOR ONE SYMBOL
                           09CD  ;
                           09CD  ; INPUTS:
                           09CD  ;
                           09CD  ;         R6      POINTS TO THE SYMBOL BLOCK
                           09CD  ;
                           09CD  ;--
                           09CD
                           09CD  MAC$PRT_SYM_INF:
            07C0 8F    BB  09CD          PUSHR   #^M<R6,R7,R8,R9,R10>
   5A    00000000'EF   D0  09D1          MOVL    L^MAC$GL_LIST_PTR,R10     ;GET CURRENT POINTER
                 5A    DD  09D8          PUSHL   R10                       ;COPY ONTO STACK
       50   04 A6      9A  09DA          MOVZBL  SYM$B_NAME(R6),R0         ; Get offset to symbol count/name
      50   56  50      C3  09DE          SUBL3   R0,R6,R0                  ; and form its address
           51  80      9A  09E2          MOVZBL  (R0)+,R1                  ; Get count and advance pointer to name
      6A   60  51      28  09E5          MOVC3   R1,(R0),(R10)             ; Copy into listing buffer
      5A   6E  0F      C1  09E9          ADDL3   #<SYM$K_TWOCOL-1>,(SP),R10 ; Point past end of name
        03 6B  2A      E1  09ED          BBC     #FLG$V_SYM2COL,(R11),5$ ; Branch if not 2 column listing
           5A  10      C0  09F1          ADDL2   #<SYM$K_MAXLEN-SYM$K_TWOCOL+1>,R10 ; Allow for 31 character symbol
                           09F4  5$:
     05 09 A6  08      E1  09F4          BBC     #SYM$V_ASN,SYM$W_FLAG(R6),10$ ;BRANCH IF NOT ASSIGNED SYMBOL
           8A  3D      90  09F9          MOVB    #^A/=/,(R10)+             ;YES--STORE EQUAL SIGN
               03      11  09FC          BRB     20$
           8A  20      90  09FE  10$:    MOVB    #^A/ /,(R10)+
           5A  09      C0  0A01  20$:    ADDL2   #9,R10                    ;MAKE ROOM FOR VALUE
      00000000'EF  5A  D0  0A04          MOVL    R10,L^MAC$GL_LIST_PTR     ;STORE CURRENT POINTER
               5A      DD  0A0B          PUSHL   R10                       ;SAVE ON STACK ALSO
     09 09 A6  00      E0  0A0D          BBS     #SYM$V_DEF,SYM$W FLAG(R6),30$ ;BRANCH IF SYMBOL DEFINED
F8 AA   08  2A  6E  00  2C  0A12          MOVC5   #0,(SP),#^A/*/,#8,-8(R10) ;NO--FILL VALUE FIELD WITH STARS
               11      11  0A19          BRB     40$
               57      D6  0A1B  30$:    INCL    R7                        ;DEFINED--STACK VALUE
  00000000'EF47  05 A6  D0  0A1D          MOVL    SYM$L_VAL(R6),L^MAC$AL_VALSTACK[R7] ;...
           50  04      9A  0A26          MOVZBL  #4,R0                     ;LIST LONGWORD VALUE
            F5D4'      30  0A29          BSBW    MAC$LIST_BYT_0            ;...
             5A 8ED0       0A2C  40$:    POPL    R1C                       ;RESTORE POINTER
       50   09 A6      3C  0A2F          MOVZWL  SYM$W_FLAG(R6),R0         ;GET SYMBOL FLAGS
        04 50  01      E1  0A33          BBC     #SYM$V_WEAK,R0,50$        ;BRANCH IF NOT WEAK
        6A   57 8F     90  0A37          MOVB    #^A/W/,0(R10)             ;YES--FLAG IT
        09 50  04      E0  0A3B  50$:    BBS     #SYM$V_ABS,R0,60$         ;BRANCH IF ABSOLUTE SYMBOL
        05 50  00      E1  0A3F          BBC     #SYM$V_DEF,R0,60$         ;NO--BRANCH IF DEFINED
                           0A43  ;
                           0A43  ; SYMBOL IS RELOCATABLE AND DEFINED
                           0A43  ;
     01 AA   52 8F     90  0A43          MOVB    #^A/R/,1(R10)             ;FLAG RELOCATABLE
        05 50  02      E1  0A48  60$:    BBC     #SYM$V_GLOBL,R0,70$       ;BRANCH IF NOT GLOBAL SYMBOL
     02 AA   47 8F     90  0A4C          MOVB    #^A/G/,2(R10)             ;YES--FLAG IT
        05 50  03      E1  0A51  70$:    BBC     #SYM$V_EXTRN,R0,80$       ;BRANCH IF NOT EXTERNAL
     03 AA   58 8F     90  0A55          MOVB    #^A/X/,3(R10)             ;YES--FLAG IT
        05 50  0A      E1  0A5A  80$:    BBC     #SYM$V_ODBG,R0,90$        ;BRANCH IF NO DEBUG OUTPUT FOR SYMBOL
     04 AA   44 8F     90  0A5E          MOVB    #^A/D/,4(R10)             ;YES--FLAG IT
           50  0D      B3  0A63  90$:    BITW    #SYM$M_DEF!SYM$M_GLOBL!SYM$M_EXTRN,R0 ;
               05      12  0A66          BNEQ    100$                      ;IF DEFINED, GLOBAL OR EXTERNAL
```

```
              05 AA   55 8F   90 0A68            MOVB    #^A/U/,5(R10)             ;NO--MARK AS UNDEFINED
                   5A   09   C0 0A6D       100$:  ADDL2   #9,R10                    ;MOVE TO PSECT SPOT
          00000000'EF   5A   D0 0A70             MOVL    R10,L^MAC$GL_LIST_PTR     ;STORE POINTER
                20 50   04   E0 0A77             BBS     #SYM$V_ABS,R0,110$        ;BRANCH IF ABSOLUTE
                50   0C A6   9A 0A7B             MOVZBL  SYM$B_SEG(R6),R0          ;GET SEGMENT NUMBER
                        0A   13 0A7F             BEQL    105$                      ;DON'T FIDDLE WITH THE ABS PSECT.
                        07   E0 0A81             BBS     #SYM$V_REF,-              ;HAS THE BLANK PSECT BEEN FILTERED OUT?
          02 00000009'GF        0A83                     G^PSECT$BLANK+SYM$W_FLAG,105$ ;BRANCH IF NOT.
                        50   D7 0A89             DECL    R0                        ;YES, ADJUST THE PSECT# IN SYMBOL TABLE.
                        57   D6 0A8B       105$:  INCL    R7                        ;STACK IT
        00000000'EF47   50   D0 0A8D             MOVL    R0,L^MAC$AL_VALSTACK[R7]  ;...
                   50   01   9A 0A95             MOVZBL  #1,R0                     ;LIST ONE BYTE
                   F565'   30 0A98             BSBW    MAC$LIST_BYT_0            ;...
      00000000'EF   8E   28   C1 0A9B       110$:  ADDL3   #40,(SP)+,L^MAC$GL_LIST_PTR ; Figure new pointer and store it
                07 6B   2A   E1 0AA3             BBC     #FLG$V_SYM2COL,(R11),120$ ; Branch if not 2 column listing
          00000000'EF   12   C0 0AA7             ADDL2   #18,L^MAC$GL_LIST_PTR     ; Allow for larger symbol name
                             0AAE       120$:
                07C0 8F   BA 0AAE             POPR    #^M<R6,R7,R8,R9,R10>      ;RESTORE REGS
                          05 0AB2             RSB
                             0AB3
                             0AB3     MAC$LIST_INIT:
        50   00000000'EF   9E 0AB3             MOVAB   L^MAC$AB_LST_END,R0       ;POINT TO BUFFER
          00000000'EF   50   D0 0ABA             MOVL    R0,L^MAC$GL_LIST_PTR      ;SET POINTER
 60   0084 8F   20   6E   00   2C 0AC1             MOVC5   #0,(SP),#^A/ /,#132,(R0);FILL BUFFER WITH BLANKS
                          05 0AC9             RSB
                             0ACA
                             0ACA             .END
```

```
AUD$K_SIZE                      = 00000010                    FLG$M_COMPEXPR                  = 00000004
BLNK                            = 00000020                    FLG$M_CONT                      = 00000008
CHR$M_COMMA_CR                  = 00000020                    FLG$M_CRF                       = 40000000
CHR$M_ILL_CHR                   = 00000040                    FLG$M_CRSEEN                    = 00000001
CHR$M_NUM_BER                   = 00000010                    FLG$M_DATRPT                    = 00000010
CHR$M_SPA_MSK                   = 00000001                    FLG$M_DBGOUT                    = 00004000
CHR$M_SYM_CH1                   = 00000008                    FLG$M_DLIMSTR                   = 00008000
CHR$M_SYM_CHR                   = 00000004                    FLG$M_ENDMCH                    = 00000020
CHR$M_SYM_DLM                   = 00000002                    FLG$M_EVALEXPR                  = 00000040
CHR$V_COMMA_CR                  = 00000005                    FLG$M_EXPOPT                    = 00000080
CHR$V_CVTLWC                    = 00000061                    FLG$M_EXTERR                    = 00010000
CHR$V_ILL_CHR                   = 00000006                    FLG$M_EXTWRN                    = 00020000
CHR$V_NOCVT                     = 0000007F                    FLG$M_FIRSTLN                   = 00000200
CHR$V_NUM_BER                   = 00000004                    FLG$M_IFSTAT                    = 00800000
CHR$V_SPA_MSK                   = 00000000                    FLG$M_IIF                       = 00400000
CHR$V_SYM_CH1                   = 00000003                    FLG$M_INSERT                    = 00000100
CHR$V_SYM_CHR                   = 00000002                    FLG$M_IRPC                      = 20000000
CHR$V_SYM_DLM                   = 00000001                    FLG$M_LEXOP                     = 00000002
CR                              = 0000000D                    FLG$M_LSTXST                    = 00000200
CREF_TREE_OUT                     0000078F R      04          FLG$M_MAC2COL                   = 00000800
CRF$K_DELETE                      ********    X   04          FLG$M_MACL                      = 00000800
CRF$L_K1FMTBL                   = 00000048                    FLG$M_MACLTB                    = 08000000
CRF$M_DEFAULT                   = 00000012                    FLG$M_MACTXT                    = 00010000
CRF$M_DIR                       = 00000001                    FLG$M_MEBLST                    = 00001000
CRF$M_MACROS                    = 00000002                    FLG$M_MOREARG                   = 00002000
CRF$M_OPCODES                   = 00000004                    FLG$M_MOREINP                   = 00000008
CRF$M_REGISTERS                 = 00000008                    FLG$M_NEWPND                    = 00000400
CRF$M_SYMBOLS                   = 00000010                    FLG$M_NOREF                     = 01000000
CRF$OUT                           ********    X   04          FLG$M_NTYPEPC                   = 00000020
CRF$V_DIR                       = 00000000                    FLG$M_NULCHR                    = 00040000
CRF$V_MACROS                    = 00000001                    FLG$M_OBJXST                    = 00200000
CRF$V_OPCODES                   = 00000002                    FLG$M_OPNDCHK                   = 00000100
CRF$V_REGISTERS                 = 00000003                    FLG$M_OPRND                     = 00002000
CRF$V_SYMBOLS                   = 00000004                    FLG$M_OPTVFLIDX                 = 00001000
DBG$C_LIT_DAT                   = 00000000                    FLG$M_ORDLST                    = 00020000
DBG$C_MEND                      = 000000BD                    FLG$M_P2                        = 00004000
DBG$C_MODULE                    = 000000BC                    FLG$M_RPTIRP                    = 10000000
DBG$C_PSECT                     = 000000B8                    FLG$M_SEQFIL                    = 02000000
DBG$C_REL_DAT                   = 00000001                    FLG$M_SKAN                      = 00008000
DBG$C_ROUTINE                   = 000000BE                    FLG$M_SPECOP                    = 00000004
DBG$C_SYMBOL                    = 000000BA                    FLG$M_SPLALL                    = 04000000
DBG$K_MEND_LEN                  = 00000002                    FLG$M_STOIMF                    = 00040000
DBG$K_MODULE_LN                 = 00000008                    FLG$M_SYM2COL                   = 00000400
DBG$K_PSECT_LEN                 = 0000000C                    FLG$M_TOCFLG                    = 00080000
DBG$K_ROUTIN_LN                 = 00000008                    FLG$M_UPAFLG                    = 00000010
DBG$K_SYMBOL_LN                 = 00000008                    FLG$M_UPDFIL                    = 00000080
ENB$G_GLOBAL                      ********    X   04          FLG$M_UPMARG                    = 00000040
ENB$G_TRACEBACK                   ********    X   04          FLG$M_XCRF                      = 80000000
EOM$C_ABORT                     = 00000003                    FLG$V_ALLCHR                    = 00000000
EOM$C_ERROR                     = 00000002                    FLG$V_BOL                       = 00000001
EOM$C_SUCCESS                   = 00000000                    FLG$V_CHKLPND                   = 00000014
EOM$C_WARNING                   = 00000001                    FLG$V_COMPEXPR                  = 00000002
FF                              = 0000000C                    FLG$V_CONT                      = 00000003
FIN_ASM_EXIT                      00000787 R      04          FLG$V_CRF                       = 0000001E
FLG$M_ALLCHR                    = 00000001                    FLG$V_CRSEEN                    = 00000020
FLG$M_BOL                       = 00000002                    FLG$V_DATRPT                    = 00000004
FLG$M_CHKLPND                   = 00100000                    FLG$V_DBGOUT                    = 0000002E
```

```
FLG$V_DLIMSTR                    = 0000002F                MAC$AB_IFP_FAO                    ********    X    04
FLG$V_ENDMCH                     = 00000005                MAC$AB_IND_HDR                    ********    X    04
FLG$V_EVALEXPR                   = 00000006                MAC$AB_IND_HDRA                   ********    X    04
FLG$V_EXPOPT                     = 00000007                MAC$AB_IND_HDRB                   ********    X    04
FLG$V_EXTERR                     = 00000030                MAC$AB_LINEBF                     ********    X    04
FLG$V_EXTWRN                     = 00000031                MAC$AB_LINE_END                   ********    X    03
FLG$V_FIRSTLN                    = 00000029                MAC$AB_LPG_FMT                    ********    X    04
FLG$V_IFSTAT                     = 00000017                MAC$AB_LST_END                    ********    X    03
FLG$V_IIF                        = 00000016                MAC$AB_MCD_FAO                    ********    X    04
FLG$V_INSERT                     = 00000008                MAC$AB_MLB_FAO                    ********    X    04
FLG$V_IRPC                       = 0000001D                MAC$AB_MLB_HD2                    ********    X    04
FLG$V_LEXOP                      = 00000021                MAC$AB_MLB_HD2A                   ********    X    04
FLG$V_LSTXST                     = 00000009                MAC$AB_MLB_HDR                    ********    X    04
FLG$V_MAC2COL                    = 0000002B                MAC$AB_MLB_HDRA                   ********    X    04
FLG$V_MACL                       = 0000000B                MAC$AB_MLB_HDRB                   ********    X    04
FLG$V_MACLTB                     = 0000001B                MAC$AB_MLB_SUM                    ********    X    04
FLG$V_MACTXT                     = 00000010                MAC$AB_MLB_TOT                    ********    X    04
FLG$V_MEBLST                     = 0000000C                MAC$AB_NOERRM                     ********    X    04
FLG$V_MOREARG                    = 0000002D                MAC$AB_OBSR_FAO                   ********    X    04
FLG$V_MOREINP                    = 00000023                MAC$AB_PSC_HD2                    ********    X    04
FLG$V_NEWPND                     = 0000000A                MAC$AB_PSC_HD2A                   ********    X    04
FLG$V_NOREF                      = 00000018                MAC$AB_PSC_HDR                    ********    X    04
FLG$V_NTYPEPC                    = 00000025                MAC$AB_PSC_HDRA                   ********    X    04
FLG$V_NULCHR                     = 00000032                MAC$AB_PSC_HDRB                   ********    X    04
FLG$V_OBJXST                     = 00000015                MAC$AB_PSS_FAO                    ********    X    04
FLG$V_OPNDCHK                    = 00000028                MAC$AB_PSS_MSG                    ********    X    04
FLG$V_OPRND                      = 0000000D                MAC$AB_RNT_MSG                    ********    X    04
FLG$V_OPTVFLIDX                  = 0000002C                MAC$AB_RUN_HDR                    ********    X    04
FLG$V_ORDLST                     = 00000011                MAC$AB_RUN_HDRA                   ********    X    04
FLG$V_P2                         = 0000000E                MAC$AB_SBT_IDNT                   ********    X    04
FLG$V_RPTIRP                     = 0000001C                MAC$AB_STB_MSG                    ********    X    04
FLG$V_SEQFIL                     = 00000019                MAC$AB_SYP_FAO                    ********    X    04
FLG$V_SKAN                       = 0000000F                MAC$AB_TITLE                      ********    X    04
FLG$V_SPECOP                     = 00000022                MAC$AB_WAS_MSG                    ********    X    04
FLG$V_SPLALL                     = 0000001A                MAC$AB_WERE_MG                    ********    X    04
FLG$V_STOIMF                     = 00000012                MAC$AB_WSL_FAO                    ********    X    04
FLG$V_SYM2COL                    = 0000002A                MAC$AL_CRF_TB1                    ********    X    04
FLG$V_TOCFLG                     = 00000013                MAC$AL_CRF_TB1A                   ********    X    04
FLG$V_UPAFLG                     = 00000024                MAC$AL_CRF_TB1B                   ********    X    04
FLG$V_UPDFIL                     = 00000027                MAC$AL_CRF_TB2                    ********    X    04
FLG$V_UPMARG                     = 00000026                MAC$AL_CRF_TB2A                   ********    X    04
FLG$V_XCRF                       = 0000001F                MAC$AL_CRF_TB2W                   ********    X    04
HASHSZ                           = 0000007F                MAC$AL_CRF_TB2WA                  ********    X    04
HYPHEN                           = 0000002D                MAC$AL_CRF_TB3                    ********    X    04
INP$K_BUFSIZ                     = 000003E8                MAC$AL_CRF_TB5                    ********    X    04
INT$K_BUFSIZ                     = 000013F4                MAC$AL_CRF_TB7                    ********    X    04
INT$K_BUFWRN                     = 00001390                MAC$AL_CRF_TB8                    ********    X    04
LST$K_BUFSIZ                     = 00000086                MAC$AL_CRF_TB9                    ********    X    04
LST$K_L_P_PAGE                   = 0000003C                MAC$AL_VALSTACK                   ********    X    04
LST$K_TITLE_SIZ                  = 00000028                MAC$AW_IND_NAMS                   ********    X    04
MAC$AB_CMD_FAO                     ********    X    04     MAC$AW_RNT_PTRS                   ********    X    04
MAC$AB_CRF_MSG                     ********    X    04     MAC$AW_TIM_PTRS                   ********    X    04
MAC$AB_CRF_TB4                     ********    X    04     MAC$CHRBYT                        ********    X    04
MAC$AB_CRF_TB6                     ********    X    04     MAC$DBG_NAM_OUT                   00000854 R      04
MAC$AB_DEF_TITL                    ********    X    04     MAC$DBG_PSECT                     00000892 R      04
MAC$AB_ERRM                        ********    X    04     MAC$DBG_VAL_OUT                   00000866 R      04
MAC$AB_FAO_TIM                     ********    X    04     MAC$DEC_OUT_L2R                   ********    X    04
```

```
MAC$FAOUTS                        ********   X    04    MERGE_LISTS                        00000062 R      04
MAC$FINISH_ASM                    000000FF RG     04    MLF$K_BLKSIZ                       00000177
MAC$GL_CMDLEN                     ********   X    04    MLF$K_RSFNLN                     = 000000FF
MAC$GL_CMDLIN                     ********   X    04    MLF$L_CTINDEX                      00000014
MAC$GL_CRF_FLG                    ********   X    04    MLF$L_MCDEF                        00000008
MAC$GL_DIRFLG                     ********   X    04    MLF$L_QLINK                        00000000
MAC$GL_ERRCT                      ********   X    04    MLF$Q_FNAMDS                       0000000C
MAC$GL_ERR_LIST                   ********   X    04    MLF$T_FNAM                         00000078
MAC$GL_INFOCNT                    ********   X    04    MLF$X_NAMBLK                       00000018
MAC$GL_INTPAGRQ                   ********   X    04    NAM$C_BLN                        = 00000060
MAC$GL_LINELN                     ********   X    04    NAM$C_MAXRSS                     = 000000FF
MAC$GL_LINE_CNT                   ********   X    04    NEW_SBT_CK_PAGE                    00000813 RG     04
MAC$GL_LINK_PTR                   ********   X    04    OBJ$C_DBG                        = 00000004
MAC$GL_LIST_LVL                   ********   X    04    OBJ$C_EOM_ABORT                  = 00000003
MAC$GL_LIST_PTR                   ********   X    04    OBJ$C_EOM_ERR                    = 00000002
MAC$GL_LN_PAGE                    ********   X    04    OBJ$C_EOM_OK                     = 00000000
MAC$GL_MCDEF                      ********   X    04    OBJ$C_EOM_WRN                    = 00000001
MAC$GL_MCPGRQ                     ********   X    04    OBJ$C_TBT                        = 00000005
MAC$GL_MLB_CNT                    ********   X    04    OBJ$K_BUFSIZ                     = 00000200
MAC$GL_MLB_GET                    ********   X    04    OPF$M_LASTOPR                    = 00002000
MAC$GL_MLB_MDF                    ********   X    04    OPF$M_OPTEXP                     = 00001000
MAC$GL_MLB_QUE                    ********   X    04    OPF$V_LASTOPR                    = 0000000D
MAC$GL_OBJ_RCNT                   ********   X    04    OPF$V_OPTEXP                     = 0000000C
MAC$GL_PSC_LIST                   ********   X    04    PRINT_SYM_TABLE                    0000023F R      04
MAC$GL_PSC_MAX                    ********   X    04    PRT_CMD_LIN                        00000752 R      04
MAC$GL_RECTYP                     ********   X    04    PRT_CROSS_REF                      000003D4 R      04
MAC$GL_SRC_LCNT                   ********   X    04    PRT_ERR_SUM                        0000066D R      04
MAC$GL_SYMPGREQ                   ********   X    04    PRT_ERR_SUM_END                    00000752 R      04
MAC$GL_SYM_LOCL                   ********   X    04    PRT_MEM_USE                        000004E1 R      04
MAC$GL_SYM_NLOC                   ********   X    04    PRT_MLB_STATS                      000005AB R      04
MAC$GL_WARNCT                     ********   X    04    PRT_PSECT_SYNOP                    00000355 R      04
MAC$GQ_RNT_CRF                    ********   X    04    PRT_RUN_TIM                        00000451 R      04
MAC$GQ_RNT_P2                     ********   X    04    PRT_SYM_END                        0000034C R      04
MAC$GQ_RNT_PSY                    ********   X    04    PRT_SYM_LOOP                       00000289 R      04
MAC$GQ_RNT_SRT                    ********   X    04    PSC$B_NAME                         00000004
MAC$GQ_RNT_SYO                    ********   X    04    PSC$B_SEG                          0000000C
MAC$GQ_RNT_TOT                    ********   X    04    PSC$B_UNUSED                       0000000B
MAC$G_ERRBFDES                    00000000 RG     03    PSC$G_OPTIONS                      ********   X    04
MAC$K_LIST_SIZE                   ********   X    04    PSC$K_BLKSIZ                       00000013
MAC$LIST_BYT_0                    ********   X    04    PSC$K_NO_OPTNS                   = 0000000A
MAC$LIST_INIT                     00000AB3 R      04    PSC$L_CURLOC                       0000000F
MAC$LST_PAG_HDR                   ********   X    04    PSC$L_LINK                         00000000
MAC$OUTOBJ                        ********   X    04    PSC$L_MAXLGTH                      00000005
MAC$OUT_LW                        ********   X    04    PSC$M_ABS                        = FFFFFFF7
MAC$PRT_SYM_INF                   000009CD R      04    PSC$M_ALIGNFLG                   = 00004000
MAC$PSECT_PRINT                   000008EA R      04    PSC$M_ALLOPTNS                   = 000003FF
MAC$SORT_TABLE                    00000000 RG     04    PSC$M_BYTE                       = 00004000
MAC$STOIM                         ********   X    04    PSC$M_CON                        = FFFFFFFB
MAC$TERM_BLANK                    ********   X    04    PSC$M_DEFAULT                    = 000001C8
MAC$TIMER_OFF                     ********   X    04    PSC$M_EXE                        = 000000C0
MAC$TIMER_ON                      ********   X    04    PSC$M_GBL                        = 00000010
MAC$WRITE_TERM                    ********   X    04    PSC$M_LCL                        = FFFFFFEF
MAC$WRTLST                        ********   X    04    PSC$M_LIB                        = 00000002
MAC$WRTOBJ                        ********   X    04    PSC$M_LONG                       = 00004800
MAC$WRT_BLNKLIN                   ********   X    04    PSC$M_NOEXE                      = FFFFFFBF
MAC$WRT_FAOUTS                    ********   X    04    PSC$M_NOPIC                      = FFFFFFFE
MAC_SUBSYS                      = 0000007D              PSC$M_NORD                       = FFFFFF7F
```

```
PSC$M_NOSHR                     = FFFFFFDF              SYM$L_VAL                     00000005
PSC$M_NOVEC                     = FFFFFDFF              SYM$M_ABS                   = 00000010
PSC$M_NOWRT                     = FFFFFEFF              SYM$M_ASN                   = 00000100
PSC$M_OVR                       = 00000004              SYM$M_CRFO                  = 00002000
PSC$M_PAGE                      = 00006400              SYM$M_DEBUG                 = 00000020
PSC$M_PIC                       = 00000001              SYM$M_DEF                   = 00000001
PSC$M_QUAD                      = 00004C00              SYM$M_DELMAC                = 00000200
PSC$M_RD                        = 00000080              SYM$M_EPT                   = 00000200
PSC$M_REL                       = 00000008              SYM$M_EXTRN                 = 00000008
PSC$M_SHR                       = 00000020              SYM$M_GLOBL                 = 00000004
PSC$M_USR                       = FFFFFFFD              SYM$M_LOCAL                 = 00000040
PSC$M_VEC                       = 00000200              SYM$M_ODBG                  = 00000400
PSC$M_WORD                      = 00004400              SYM$M_REF                   = 00000080
PSC$M_WRT                       = 00000180              SYM$M_RELPSECT              = 00000800
PSC$S_ALIGNMENT                 = 00000004              SYM$M_SUPR                  = 00004000
PSC$V_ALIGNFLG                  = 0000000E              SYM$M_WEAK                  = 00000002
PSC$V_ALIGNMENT                 = 0000000A              SYM$M_XCRF                  = 00001000
PSC$V_EXE                       = 00000006              SYM$V_ABS                   = 00000004
PSC$V_GBL                       = 00000004              SYM$V_ASN                   = 00000008
PSC$V_LIB                       = 00000001              SYM$V_CRFO                  = 0000000D
PSC$V_OVR                       = 00000002              SYM$V_DEBUG                 = 00000005
PSC$V_PIC                       = 00000000              SYM$V_DEF                   = 00000000
PSC$V_RD                        = 00000007              SYM$V_DELMAC                = 00000009
PSC$V_REL                       = 00000003              SYM$V_EPT                   = 00000009
PSC$V_SHR                       = 00000005              SYM$V_EXTRN                 = 00000003
PSC$V_VEC                       = 00000009              SYM$V_GLOBL                 = 00000002
PSC$V_WRT                       = 00000008              SYM$V_LOCAL                 = 00000006
PSC$W_FLAG                        00000009              SYM$V_ODBG                  = 0000000A
PSC$W_OPTIONS                     0000000D              SYM$V_REF                   = 00000007
PSECT$BLANK                       ********   X    04    SYM$V_RELPSECT              = 0000000B
PSECT$MAIN                        ********   X    04    SYM$V_SUPR                  = 0000000E
PSECT_OPT_MATCH                   0000097F R      04    SYM$V_WEAK                  = 00000001
RDX$V_BINARY                    = 00000000              SYM$V_XCRF                  = 0000000C
RDX$V_DECIMAL                   = 00000002              SYM$W_FLAG                    00000009
RDX$V_DOUBLE                    = 00000005              SYS$ADJWSL                    ********   GX   04
RDX$V_FLOAT                     = 00000004              SYS$FAOL                      ********   GX   04
RDX$V_GFLOAT                    = 00000006              TAB                         = 00000009
RDX$V_HEX                       = 00000003              TIR$C_STA_PB                = 00000004
RDX$V_HFLOAT                    = 00000007              TIR$C_STA_PL                = 00000006
RDX$V_OCTAL                     = 00000001              TIR$C_STO_L                 = 00000016
REG$_PC                         = 0000000F              TIR$C_STO_LW                = 00000016
SEMI                            = 0000003B              TIR$C_STO_PIDR              = 0000001B
SORT_EXIT                         000000F5 R      04    X1                          = 00000033
SORT_LISTS                        00000027 R      04    X2                          = 00080000
STB$K_PG_MISS                   = 0000000A
SYM$B_NAME                        00000004
SYM$B_SEG                         0000000C
SYM$B_TOKEN                       0000000B
SYM$F_DEF                       = 00000002
SYM$F_REL                       = 00000008
SYM$F_UNI                       = 00000004
SYM$F_VALIDATE                  = 00000010
SYM$F_WEAK                      = 00000001
SYM$K_BLKSIZ                      0000000D
SYM$K_MAXLEN                    = 0000001F
SYM$K_TWOCOL                    = 00000010
SYM$L_LINK                        00000000
```

```
                                                              H 10
MAC$FINISH                      ROUTINES FOR FINISHING ASSEMBLY          16-SEP-1984 02:15:54  VAX/VMS Macro V04-00        Page  32
Psect synopsis                                                            5-SEP-1984 01:48:10  [MACRO.SRC]FINISH.MAR;1           (19)

                                             +----------------+
                                             ! Psect synopsis !
                                             +----------------+

PSECT name                      Allocation           PSECT No.  Attributes
----------                      ----------           ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (   0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
. BLANK .                       00000000  (     0.)  01 (   1.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$ABS$                           00000177  (   375.)  02 (   2.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
MAC$RW_DATA                     00000008  (     8.)  03 (   3.)  NOPIC    USR    CON    REL    LCL NOSHR NOEXE   RD     WRT NOVEC LONG
MAC$RO_CODE_P3                  00000ACA  (  2762.)  04 (   4.)  NOPIC    USR    CON    REL    GBL NOSHR   EXE   RD   NOWRT NOVEC LONG

                                           +------------------------+
                                           ! Performance indicators !
                                           +------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     29    00:00:00.07    00:00:00.96
Command processing                106    00:00:00.35    00:00:03.59
Pass 1                            428    00:00:10.26    00:00:43.38
Symbol table sort                   0    00:00:01.28    00:00:05.53
Pass 2                            237    00:00:02.65    00:00:13.49
Symbol table output                51    00:00:00.25    00:00:01.32
Psect synopsis output               2    00:00:00.02    00:00:00.01
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals              855    00:00:14.88    00:01:08.29

The working set limit was 1800 pages.
88811 bytes (174 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1272 non-local and 115 local symbols.
1165 source lines were read in Pass 1, producing 36 object records in Pass 2.
43 pages of virtual memory were used to define 42 macros.

                                          +--------------------------+
                                          ! Macro library statistics !
                                          +--------------------------+

Macro library name                        Macros defined
------------------                        --------------
_$255$DUA28:[MACRO.OBJ]MACRO.MLB;1                     8
_$255$DUA28:[SYSLIB]STARLET.MLB;2                     10
TOTALS (all libraries)                                18

1364 GETS were required to define 18 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:FINISH/OBJ=OBJ$:FINISH MSRC$:FINISH/UPDATE=(ENH$:FINISH)+LIB$:MACRO/LIB
```

```
MA
Sy

IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
IN
KA
KA
KA
KA
KA
KA
KB
KB
KE
KE
KE
KE
KE
KE
KE
KE
KE
KC
KD
KD
```

0225 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DATA
LIS

DEFINE
LIS

FLOAT
LIS

ERRMSG
LIS

GETARG
LIS

DATA1
LIS

INPUT
LIS

ERROR
LIS

FINISH
LIS

GETCMD
LIS